Anno IX - N. 78 - 25 ottobre - Sped. Abb. Post. Gr III/70 - CR - Distr.: Parrini

SPECIALE

Riscrivi autocad in basic

C 64/128

- *A caccia di sprite*
- *Effetti video*
- *Mini grafica*

AMIGA

- *SYS 49152 anche su Amiga*
- *I comandi Amiga dos*
- *Accessori per A 500*

SPECIALE

C x C=

Corso completo di linguaggio "C" per C64, Amiga e PC

Leggo VR perché mi dà la rotta

Il lettore di VR è giovane, dinamico, creativo. Di cultura e reddito superiore alla media, possiede spesso più di un videoregistratore, oltre all'impianto hi-fi e al computer: nel tempo libero, non rinuncia ai viaggi in Italia e all'estero, e a cinema, teatro e spettacoli sportivi in genere. Usa il videoregistratore non solo per i programmi tv o preincisi, ma anche per riprendere i momenti felici in famiglia, per creare una videoteca personale.
E tu, che tipo di lettore sei?

VR
VIDEOREGISTRARE

IMMAGINE MARSILII

Sommario

Amigames

Da pagina **41** a pagina **56** la consueta rassegna dei videogames in arrivo; ed i relativi commenti, severi come al solito!

Campus 64 / 128

Campus Amiga

Usa il tuo computer

Rubriche

E' bene ricordare che...

Parleremo del **C/128** solo fino alla fine del 1990.
Dal **gennaio '91** verrà evasa prevalentemente la corrispondenza pervenuta a mezzo BBS (modem).
Dal **gennaio '91** verranno privilegiate le collaborazioni che perverranno in Redazione a mezzo BBS.
Tra breve verrà dato ampio spazio, oltre che ad **Amiga**, anche al favoloso mondo **Ms-Dos**! (ma già lo stiamo facendo...)
Dal **dicembre '91** non verranno più affrontati argomenti relativi al C/64.
La Scuola è vicina...

Commodore Computer Club

Direttore: Alessandro de Simone
Coordinatore: Marco Miotti

Redazione / Collaboratori:
Davide Ardizzone - Claudio Baiocchi
Luigi Callegari - Umbero Colapicchioni
Donato De Luca - Carlo D'Ippolito
Valerio Ferri - Michele Maggi
Giancarlo Mariani - Domenico Pavone
Armando Sforzi - Dario Pistella
Fabio Sorgato- Valentino Spataro
Franco Rodella- Stefano Simonelli
Luca Viola

Grafica: Arturo Ciaglia

Direzione:
Via Mosè, 22 cap. 20090 OPERA (Mi)

Telefono 02 / 55.50.03.10
Fax 02 / 57.60.30.39
BBS 02 / 57605211

Pubblicità:
Leandro Nencioni (dir. vendite)
Via Mosè, 22 20090 Opera (Mi)
tel. 02 / 55.50.03.10

Emilia Romagna:
Spazio E
P.zza Roosvelt, 4 cap. 40123 Bologna
Tel. 051 / 23.69.79

Toscana, Marche, Umbria
Mercurio s.r.l. Via Rodari, 9
S. G.nni Valdarno (Ar)
Tel. 055 / 94.74.44

Lazio, Campania
Spazio Nuovo
Via P. Foscari, 70
cap. 00139 Roma
tel. 06 / 81.09.679

Abbonamenti: Liliana Spina
Arretrati e s/w: Lucia Dominoni

Tariffe: Prezzo per copia L. 6000
Abbonamento annuo (11 fascicoli) L. 50000
Estero: L. 100000 - Indirizzare versamenti a:
Systems Editoriale Srl c/c 37952207 oppure
inviare comune assegno bancario non
trasferibile e barrato due volte a:
Systems Editoriale Srl (servizio arretrati)
Via Mosè, 22
cap. 20090 OPERA (Mi)

Composizione: Systems Editoriale
La Litografica Srl Busto Arsizio (Va)

Registrazioni: Tribunale di Milano
n. 370 del 2/10/82

Direttore Responsabile: Michele Di Pisa

Spedizioni in abbonamento postale gruppo
III. Pubblicità inferiore al 70%

Distributore: Parrini - Milano

Periodici Systems: Banca Oggi -
Commodore Club (disco) - Commodore
Computer Club - Commodore Computer
Club (disco, produzione tedesca) - Computer
- Computer disco - Electronic Mass Media
Age - Energy Manager - Hospital
Management - Jonathan - Nursing '90 - PC
Programm (disco) - Personal Computer -
Security - Software Club (cassetta ed.
italiana) - TuttoGatto - Videoteca
VR Videoregistrare

Una scuola in più

Da qualche mese sulla nostra rivista viene compiuta un'operazione, spesso dolorosa, che vede impegnati in prima persona i nostri lettori.

A questi è stato dapprima suggerito, quindi "imposto", di cambiar computer; oppure hobby. La brutalità del nostro atteggiamento non ha assolutamente, come secondo fine, quello di incrementare il fatturato di questa o quell'azienda. Al contrario, la necessità di cambiar computer non è altro che un voler / dover restare al passo con i tempi, mutevoli con incessante rapidità.

La stragrande maggioranza dei lettori ha compreso che, nel valutare la spesa necessaria per il gran passo, sono molti i parametri che entrano in giuoco per "indorare la pillola"; non ultima la considerazione che, ai tempi dell'esplosione del Boom informatico, un **C/64** costava, al pubblico, l'**intero stipendio mensile** netto di un impiegato.

Oggi l'inflazione, da una parte, ed il progresso tecnologico, dall'altra, consentono di proporre un **Amiga** (o un sistema **Ms-Dos** equivalente dal punto dell'hardware) a poco più della **metà dello stipendio mensile** dello stesso impiegato di pari livello.

Comunque sia, il mondo sta andando verso il 16 / 32 bit e non si può restare ancorati al passato.

Lo dimostra perfino la Scuola Italiana che, lanciando il PIANO NAZIONALE DELL'INFORMATICA, costringerà (è il termine adatto) ad usare i computers non solo gli inseganti di matematica, ma tutti i docenti e, di conseguenza, tutti gli allievi degli istituti di ogni ordine e grado.

Da qualche tempo giungono in Redazione lettere e telefonate di docenti che, sensibili all'aggiornamento professionale, chiedono consigli sul computer da acquistare; altri, già esperti, propongono di collaborare su temi, inutile dirlo, di sapore scolastico.

A tutti, quindi, lanciamo un messaggio da queste pagine:

Insegnanti ed alunni, tenetevi pronti; stiamo per lanciare una nuova iniziativa che interesserà il mondo della scuola.

E, trattandosi di didattica, dovrà trattarsi, come abitudine della nostra testata, di qualcosa di valido; immodestia a parte...

Alessandro de Simone

LA VOSTRA POSTA

(a cura di A. de Simone)

Errore concorrenziale

Trascrivendo un listato per Amiga 500 (da una rivista vostra concorrente) compare la segnalazione di errore "Subscript Out Of Range" evidenziando il rigo:
If X < 128 Then Square%(X) = 127 Else Sqare%(X) = -128
Eppure il rigo che ho trascritto è identico a quello pubblicato. Potete rispondermi voi, dal momento che la rivista citata, da me interpellata in precedenza, fa orecchio da mercante?
(Andrea T.)

L'errore non risiede nella riga indicata, bensì nel valore di **X** indicato tra le parentesi della variabile relativa al vettore di numeri interi **Square%()**.

La sintassi Square%(X) presuppone, infatti, che il valore X sia compatibile con quanto dichiarato in precedenza, di solito un'istruzione **Dim** che, infatti, è presente proprio nella prima riga del listato inviato dal nostro lettore.

Se non viene effettuata alcuna dichiarazione, l'interprete assegna, automaticamente, **11** "celle" numerate da **0** a **10**.

In conclusione, è probabile che, nella prima riga, il lettore abbia trascritto, per errore...

Dim square%(**25**)

...o simili, e non **255**; oppure che abbia digitato male la variabile (es. **sqare** e non **square**). In ogni caso è un errore di distrazione dal momento che la sintassi risulta corretta.

Il motivo per cui la rivista "concorrente" non abbia risposto al quesito va ricercato, quindi, nella notevole banalità del problema posto, che poteva essere risolto leggendo con attenzione il manuale di Amiga.

Perchè mai, vi chiederete, abbiamo risposto noi? La nostra infinita bontà ci obbliga a questo ed altro...

Semplice lampeggio

Ho esaminato il programma in linguaggio macchina "Un messaggio lampeggiante" (vedi C.C.C. n. 75), ma ritengo che il Basic sia sufficiente per raggiungere lo stesso scopo; anzi, è necessario inserire una ciclo di ritardo per evitare un'eccessiva velocità di elaborazione.
(Ivan Tibaldi - Monza)

Il nostro lettore, che ha inviato il seguente listato...

```
100 rem lampeggio righe
(c/64)
110 poke 53280,0: poke
53281,0
120 print chr$(147): dim a(5)
130 for i=1 to 5: read a(i):
next
140 for i=1 to 5: poke 646,
a(i)
150 gosub 190: next
160 for i=5 to 1 step -1
170 poke 646, a(i): gosub
190
180 next: goto 140
190 print chr$(19) "prima ri-
ga"
200 print "seconda riga"
210 print "terza  riga"
220 print "quarta  riga"
230 print "quinta  riga"
240 print "sesta  riga"
250 for k=1 to 20: next: re-
turn
260 data 0, 11, 12, 15, 1
```

...ha perfettamente ragione.

Egli, infatti, ricorre ad un trucchetto davvero intelligen-

te che sfrutta la potenzialità offerta dalla **locazione 646**, incaricata di indicare il colore corrente del cursore.

A onor del vero c'è da dire che è possibile accorgersi, anche con bassi valori del ciclo di ritardo, della successione di stampa delle singole righe. Ciò non si verifica, invece, con il programma in linguaggio macchina pubblicato sul n. 75 che, grazie alla notevole velocità di elaborazione, rende praticamente istantanea la visualizzazione delle quattro righe di schermo.

Il nostro lettore, tuttavia, dimostra che, con un po' di fantasia, è certamente possibile sfruttare al massimo linguaggi di programmazione (come il Basic) che, a torto, vengono spesso snobbati da utenti "esperti".

Modifiche al modem

Sono in possesso di un modem Taihaho(?) che collegavo alla porta utente del C/64. Dal momento che sono passato ad Amiga mi piacerebbe utilizzarlo per il mio nuovo computer. Quali sono le connessioni da effettuare?

(Roberto Cavazza - Bologna)

Non conosciamo il modem in questione, ma non penso che il gioco valga la candela. Si tratta, quasi certamente, di un modem dalle limitate potenzialità che quasi certamente non dispone della compatibilità richiesta dai pacchetti di comunicazione disponibili per Amiga (per non parlare della velocità di rice/trasmissione dei dati). Potrebbero, inoltre, presentarsi problemi dovuti all'alimentazione dell'apparecchio.

Il nostro consiglio, quindi, non può che essere uno, ed uno solo: vendere il vecchio modem ad un irriducibile 64-sta (di simili individui, purtroppo, ce ne sono ancora) ed acquistare un modem degno del tuo fiammante Amiga.

Politica e Telematica

A titolo di pura curiosità, riportiamo una notizia capitata in Redazione per caso. Il Partito Radicale (quello dell'imprevedibile Pannella e della Cicciottina, per intenderci) ha attivato un servizio telematico che sembra offrire numerosi "servizi". Consente agli abbonati, infatti, di partecipare a conferenze elettroniche, consultare archivi e scambiarsi messaggi personali. Ci si può collegare ad **Agorà** (è questo il nome del servizio telematico) attraverso la linea telefonica normale (tel. 06-68.92.82.8, dotata di ben 10 linee a ricerca automatica; parametri 300 - 1200 - 2400 baud N81) oppure attraverso la rete a pacchetto Itapac (NUA 26500016 - 32 canali - 1200 baud N81). Per maggiori informazioni, comunque, ci si può rivolgere telefonando "a voce" al numero 06-68.64.233 oppure 06- 68.48.119.

Perchè riportiamo questa notizia? Semplicemente perchè ci sembra doveroso segnalare lo spirito di modernità che inizia ad aleggiare nelle borboniche strutture dei polverosi palazzi politici. Chi è in grado di segnalare analoghe iniziative, sostenute da altri partiti, potrà scrivere in Redazione specificando le modalità di collegamento alle loro banche dati.

Campioni del software

Come abbiamo già affermato in un numero precedente, è stata presa la decisione di non pubblicare le soluzioni dei videogames che risultano copiate da altre testate.

Per motivi analoghi, inoltre, non possiamo pubblicare le Poke relative a vite infinite, cambi di schermate (ed altre tecniche di manipolazione del gioco), se il lettore non indica chiaramente le modalità che ha seguito per giungere alla loro individuazione.

Vengono quindi cestinate le lettere che, senza alcuna spiegazione, riportano sequenze di Poke e di Sys prive di qualsiasi commento.

Come intuitivo, non possono esser prese in considerazione nemmeno le procedure che, ricorrendo a tecniche che richiedono l'utilizzo di cartucce, programmi commerciali o di pubblico dominio, non sono altro che parte del manuale di istruzioni di dette procedure.

In ogni caso, anche per la rubrica "Campioni del Software" vale la regola cui debbono attenersi tutti gli aspiranti collaboratori: è indispensabile contattare, telefonicamente, il direttore di C.C.C. in persona prima di inviare il lavoro; questo rischierebbe, in caso contrario, di esser cestinato senza pietà.

Segnaliamo, comunque, il lettore Eugenio Gori C.so Italia, 48 cap 57027 San Vincenzo (Li) che, per un sofisticato lavoro di manipolazione software effettuato su numerosi giochi commercializzati, merita di essere elogiato, tanto più che, come sembra, è a capo di un piccolo club di Hackers operante nella sua città.

Rubar files

Vi sembra giusto che un lettore dotato di modem possa "rubacchiare" liberamente files e programmi dalla vostra BBS mentre chi, come me, non dispone di tale apparecchio debba esser costretto a comprare i dischetti?

(Luca Mineo - Palermo)

Anzitutto, non è vero che il servizio sia totalmente gratuito: non dimentichiamo che, alla fine di ogni bimestre, la Sip emette una bolletta, spesso salata. Inoltre è indispensabile tener conto che la tecnologia avanza. Se qualche anno fa un modem era un **lusso** per pochi, oggi rappresenta, al contrario, un utile investimento (cioè un **risparmio**). Tra breve perfino in Italia (per restare in tema di telecomunicazione) sarà possibile disporre di un radiotelefono a basso costo; come avviene, da molti lustri, nel resto del mondo civilizzato.

BBS e C/64

Spesso riesco a collegarmi con la vostra BBS solo con notevole difficoltà...

(Alberto Brambilla - Concorezzo)

La lamentela è simile a quella di altri lettori che, ostinandosi ad usare il **6499**, pretendono di avere un servizio di prima qualità.

Non inviate francobolli

La Redazione ricorda, soprattutto ai neo-lettori, che non viene mai data risposta privata ai loro quesiti, nemmeno nel caso in cui viene allegata l'affrancatura postale per la risposta. L'unico modo per ottenere informazioni è quello di documentarsi attraverso le pagine della nostra rivista, oppure di telefonare attenendosi alle precisazioni riportate spesso sulle nostre pagine.

Alcuni modem di marca semisconosciuta, poi, creano problemi perfino al nostro computer dedicato che, in certi casi, addirittura si inchioda in fase di "colloquio".

Sembra che la bassa velocità di trasmissione (solo **300 baud**) sia la causa di simili inconvenienti. Quali i rimedi suggeriti? Andiamo, non abbiamo voglia di ripeterci all'infinito...

Masochismi

Posseggo sia un C/128-D sia un sistema Ms-Dos dotato di Hard Disk ai quali, alternativamente, collego una stampante Mps -1230, provvista di entrambi gli standard di comunicazione.

Mentre, però, in ambiente Ms-Dos la compatibilità grafica è totale e completa, altrettanto non posso dire se collego la stampante con il C/128-D sul quale faccio girare i programmi Calc Result oppure Art Studio. Potete dirmi il motivo di tali malfunzionamenti?

(Attilio Scolari - S. Martino)
(Giacinto Colafatello - Biella)

Non tutti i programmi commercializzati per il C/64 (ed il C/128) girano su tutte le stampanti 803-compatibili. A volte capitano inspiegabili incompatibilità a dispetto del corretto settaggio dei parametri e dei numerosi switch presenti sulle stampanti. Diversi lettori, tuttavia, lamentano problemi collegando la 1230 al C/64 o al C/128.

Ma, Santo Cielo!, perchè mai torturarsi ed incaponirsi ad usare i **lentissimi** programmi Calc Result e Art Studio? Se si disponesse del solo C/128, capirei; ma possedendo un **signor** computer (e mi riferisco all'Ms-Dos) non riesco davvero a capire per quale motivo insistere ad impiegare un tempo almeno dieci volte superiore per svolgere funzioni che, con un elaboratore Ms-Dos, si svolgono in un battibaleno. Per ciò che riguarda il tabellone elettronico (spreadsheet) puoi usare il fantastico (e, soprattutto velocissimo e super-grafico) **Quattro** della **Borland**. Tale programma, anche se ti procuri una vecchia versione, non ti farà certo rimpingere gli angusti limiti offerti da Calc Result.

Per ciò che riguarda i programmi grafici, invece, la scelta è ancora più vasta di quella offerta dall'obsoleto sistema C/64-128.

Tabella 6510

Il fascicolo "Linguaggio Macchina e Routines Grafiche" non comprende le tabelle relative ai codici macchina del 6510. Le avete, forse, pubblicate su C.C.C?

(Alessandro Bortolussi - S. Vito)

Sul n. 40 del nostro periodico è riportato un paginone centrale in cui sono annotate tutte le informazioni relative alle singole istruzioni disponibili sui microprocessori della "famiglia" **65XX** tra cui, quindi, anche il 6510.

Puoi richedere il fascicolo in oggetto al nostro servizio arretrati.

Easy Script laureando

Vorrei usare il mio C/64, corredato del word processor Easy Script e della stampante Mps-1230, per scrivere la mia tesi di laurea in francese.

Come è possibile evidenziare i caratteri speciali (accenti e così via) di questa lingua e come attivare i vari modi di stampa della Mps-1230 dal momento che il manuale in mio possesso è in tedesco?

(Roberto Scardino - La Spezia)

Purtroppo l'inconveniente lamentato dal nostro lettore è prassi normale per chi acquista i prodotti Commodore non importati direttamente dalla Commodore Italiana, che è l'unica autorizzata a distribuire i manuali in lingua italiana.

Anche per quanto riguarda l'uso corretto di **Easy Script** sono costretto a consigliare l'acquisto del prodotto originale (in lingua italiana, mi raccomando!) oppure la rilettura dell'inserto speciale pubblicato sul **n. 41** della nostra pubblicazione, interamente dedicato al word processor più popolare per il C/64.

Obsolescenza

A causa della perdita di interesse per i "piccoli" sistemi Commodore, la Redazione di C.C.C. cede, per inutilizzo, la **tavoletta** grafica originale **Koala** (corredata di **cartuccia**, altrettanto originale) al prezzo simbolico di L. 90000. Telefonare il giovedì pomeriggio alla Redazione

Delusione per il C/128

L'anno scorso ho acquistato, presso un Commodore Point, un C/128 fornito di manuale in inglese, con la promessa che, in seguito, mi avrebbero spedito l'edizione in italiano. Dopo alcuni mesi, e numerose insistenze da parte mia, mi fu comunicato che il manuale non era più disponibile. A parte le ovvie considerazioni sulla serietà commerciale, vi chiedo di pubblicare il mio annuncio, relativo alla

Totocalcio ed Emulatore Turbo Pascal 64

Alcuni lettori hanno difficoltà ad usare due programmi della **Systems Editoriale** pubblicati, su disco, tempo addietro. Per ciò che riguarda l'emulatore Turbo Pascal è comparso un articolo di delucidazioni sul n. **64**; per lo "Speciale Totocalcio", invece, la risposta ad un lettore (apparsa nella rubrica "La vostra posta" del n. **62**) chiarisce alcuni dubbi sul corretto uso dei programmi contenuti nel dischetto.

SYSTEMS EDITORIALE PER TE

La voce
Aggiunge al C/64 nuovi comandi Basic che consentono sia di far parlare il computer, sia di farlo Cantare! Diversi esempi allegati.

Cassetta: L. 12000 - Disco: L. 15000

Raffaello
Un programma completo per disegnare, a colori, con il C/64: linee, cerchi, quadrati, eccetera. Valido sia per disegno a mano libera che geometrico.

Cassetta: L. 10000

Oroscopo
Devi solo digitare la data di nascita e le coordinate geografiche del luogo che ti ha dato i natali. Vengono quindi elaborate le varie informazioni (case, influenze dei segni astrali, eccetera) e visualizzato un profilo del tuo carattere. Valido per qualsiasi anno, è indicato sia agli esperti sia ai meno introdotti. E' allegata una tabella delle coordinate delle più note città italiane e l'elenco delle ore legali in Italia dal 1916 al 1978.

Cassetta: L. 12000 - Disco: L. 12000

Computer Music
Cassetta contenente numerosi brani di successo da far eseguire, in interrupt, al tuo C/64 sfruttando, fino in fondo, il suo generatore sonoro (SID).

Cassetta: L. 12000

Gestione Familiare
Il più noto ed economico programma per controllare le spese e i guadagni di una famiglia.

Cassetta: L. 10000 - Disco: L. 10000

Banca Dati
Il più noto ed economico programma per gestire dati di qualsiasi natura.

Cassetta: L. 10000 - Disco: L. 10000

Matematica finanziaria
Un programma completo per la soluzione dei più frequenti problemi del settore.

Cassetta: L. 10000 - Disco: L. 20000

Analisi di bilancio
Uno strumento efficace per determinare con precisione i calcoli necessari ad un corretto bilancio.

Cassetta: L. 10000 - Disco: L. 20000

Corso di Basic
Confezione contenente quattro cassette per imparare velocemente le caratteristiche delle istruzioni Basic del C/64 e i rudimenti di programmazione. Interattivo.

Cassetta: L. 19000

Corso di Assembler
Un corso completo su cassetta per chi ha deciso di abbandonare il Basic del C/64 per addentrarsi nello studio delle potenzialità del microprocessore 6502. Interattivo.

Cassetta: L. 10000

Logo Systems
Il linguaggio più facile ed intuitivo esistente nel campo dell'informatica; ideale per far avvicinare i bambini al calcolatore.
Diversi esempi allegati.

Cassetta: L. 6500

Compilatore Grafico Matematico
Uno straordinario programma compilatore, di uso semplicissimo, che permette di tracciare, sul C/64, grafici matematici Hi-Res ad altissima velocità. Esempi d'uso allegati.

Cassetta: L. 8000

Emulatore Ms-Dos e Gw-Basic
Un prodotto, unico nel suo genere, che permette di usare, sul C/64 dotato di drive, la sintassi tipica del più diffuso sistema operativo del mondo. Ideale per studenti.

Solo su disco: L. 20000

Emulatore Turbo Pascal 64
Permette di usare le più importanti forme sintattiche del linguaggio Turbo Pascal (anche grafiche!) usando un semplice C/64 dotato di drive.
Ideale per studenti.

Disco: L. 19000

Speciale drive
Questo speciale fascicolo costituisce una guida di riferimento per le unità a disco del C64/128.
Comprende anche un velocissimo turbo-disk più la mappa completa della memoria del drive.

Fascicolo + disco: L. 12000

Utility 1
Un dischetto pieno zeppo di programmi speciali per chi opera frequentemente con il drive.

Disco: L. 12000

Utility 2
Seconda raccolta di utility indispensabili per realizzare sofisticate procedure di programmazione.

Disco: L. 15000

Graphic Expander 128
Per usare il C/128 (in modo 128 e su 80 colonne) in modo grafico Hi-res. Aggiunge nuove, potenti istruzioni Basic per disegnare in Hi-Res con la massima velocità in modalità 80 colonne.

Disco: L. 27000

Directory
Come è noto, a partire dal N. 10 di "Software Club" (la rivista su disco per l'utente dei "piccoli" computer Commodore), vengono riportati tutti i listati, in formato C/64-C/128, pubblicati su "Commodore Computer Club".
In precedenza tali listati venivano inseriti, mensilmente, in un dischetto, di nome "Directory", che oltre ai programmi di C.C.C. ospitava decine di altri file tra cui musiche nell'interrupt, giochi, listati inviati dai lettori e altro.
Ogni disco, dal prezzo irrisorio, contiene quindi una vera miniera di software. Ordinando i dischetti di "Directory" si tenga conto che al N. 1 corrispondeva il contenuto del N. 34 di "Commodore Computer Club", al N. 2 il N. 35 e così via.

Ogni dischetto: L. 10000

Super Tot '64
La nuova e completa edizione del programma Tot 13 con tutti i sistemi di riduzione e di condizionamento.
Ampia sezione dedicata alla teoria.

fascicolo + disco: L. 15000

Amiga Totospeed
Finalmente anche per Amiga un programma orientato alla compilazione delle schedine totocalcio.
Fai tredici conil tuo Amiga.

disco: L. 20000

SYSTEMS EDITORIALE PER TE

Disk'o'teca
Grazie a questa nutrita raccolta di brani musicali potrete divertirvi ascoltando i migliori brani prodotti dai vostri beniamini, oltre a una serie di composizioni prodotte "in casa".
In omaggio un bellissimo poster di Sting.
Disco: L.15.000

Assaggio di primavera
Esclusivo!
In un'unica confezione potrete trovare ben due cassette di videogiochi assieme a un comodo e funzionale joystick.
Cassette: L. 15.000

LIBRI TASCABILI

64 programmi per il C/64
Raccolta di programmi (giochi e utilità) semplici da digitare e da usare. Ideale per i principianti. **(126 pag.)**
L. 4800

I miei amici C/16 e Plus/4
Il volumetto, di facile apprendimento, rappresenta un vero e proprio mini-corso di Basic per i due computer Commodore. Numerosi programmi, di immediata digitazione, completano la parte teorica. **(127 pag.)**
L. 7000

62 programmi per C/16, Plus/4
Raccolta di numerosi programmi, molto brevi e semplici da digitare, per conoscere più a fondo il proprio elaboratore.
Ideale per i principianti. **(127 pag.)**
L. 6500

Micro Pascal 64
Descrizione accurata della sintassi usata dal linguaggio Pascal "classico". Completa il volume un programma di emulazione del PL/O sia in formato Microsoft sia in versione C/64 (da chiedere, a parte, su disco). **(125 pag.)**
L. 7000

Dal registratore al Drive
Esame accurato delle istruzioni relative alle due più popolari periferiche del C/64. Diversi programmi applicativi ed esempi d'uso. (94 pag.)
L. 7000

Il linguaggio Pascal
Esame approfondito della sintassi usata nel famoso compilatore. (112 pag.)
L. 5000

Simulazioni e test per la didattica
Raccolta di numerosi programmi che approfondiscono e tendono a completare la trattazione già affrontata sul precedente volume. **(127 pag.)**
L. 7000

Dizionario dell'Informatica
Dizionario inglese-italiano di tutti i termini usati nell'informatica. (Edizione completa). **(385 pag.)**
L. 10000

Word processing: istruzioni per l'uso
Raccolta delle principali istruzioni dei più diffusi programmi di w/p per i sistemi Ms-Dos: Word-Star, Samna, Multimate Advantage, Word 3. **(79 pag.)**
L. 5000

Unix
Un volumetto per saperne di più sul sistema operativo professionale per eccellenza.
Un necessario compendio per l'utente sia avanzato che inesperto (**91 pag.**)
L. 5000

ABBONAMENTO

Commodore Computer Club
11 fascicoli: L. 50.000

ARRETRATI

Ciascun numero arretrato di C.C.C. L. 6.000

Come richiedere i prodotti Systems

Coloro che desiderano procurarsi i prodotti della Systems Editoriale devono inviare, oltre alla cifra risultante dalla somma dei singoli prodotti, L. 3500 per spese di imballo e spedizione, oppure L. 6000 se si desidera la spedizione per mezzo raccomandata.
Le spese di imballo e spedizione sono a carico della Systems se ciascun ordine è pari ad almeno L. 50000.
Per gli ordini, compilare un normale modulo di C/C postale indirizzato a:

C/C Postale N. 37 95 22 07
Systems Editoriale Srl
Via Mosè, 22
20090 Opera (MI)

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento"), non solo il vostro nominativo completo di recapito telefonico, ma anche i prodotti desiderati ed il tipo di spedizione da effettuare.
Per sveltire la procedura di spedizione sarebbe opportuno inviare, a parte, una lettera riassuntiva dell'ordine effettuato, allegando una fotocopia della ricevuta del versamento.
Chi volesse ricevere più celermente la confezione deve inviare la somma richiesta mediante assegno circolare, oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a:

Systems Editoriale
Milano

A scuola di Turbo Pascal

Molti utenti di sistemi **Ms-Dos** chiedono consigli sull'acquisto di un volume in grado di insegnare, con estrema semplicità, i primi rudimenti del linguaggio Turbo Pascal (originale Borland). In effetti il manuale, fornito a corredo del potente linguaggio, sembra rivolgersi a chi già conosce la versatile sintassi e, a tratti, non risulta del tutto comprensibile a chi compie i primi passi nel mondo della programmazione.

Ci permettiamo, quindi, di suggerire ai nostri lettori di procurarsi i libri normalmente adottati nelle scuole tecniche; questi volumi, proprio perchè destinati ad un pubblico giovane ed inesperto (e non sempre desideroso di imparare...) trattano i numerosi argomenti in modo estrememente efficace, semplice e graduale, servendosi di numerosissimi esempi di breve digitazione.

In particolare, ci permettiamo di segnalare il testo di **A. Brusamolin e V. Mantovani "Laboratorio di Informatica (Programmazione in Turbo Pascal)" edizioni CEDAM.**

Al prezzo di poco più di **20mila lire**, infatti, le **400** pagine offrono, anche al principiante più sprovveduto, la possibilità di apprendere i rudimenti del linguaggio, dal trattamento delle variabili alla elaborazione grafica di funzioni matematiche.

Il volume, ovviamente, è disponibile presso tutte le librerie specializzate in testi scolastici.

Per ciò che riguarda la **chimica**, spesso trascurata in termini di didattica informatica, la casa editrice **Masson** propone il volume **"Chimica fisica - Problemi al computer" (di Marseglia)** che illustra, in modo pratico, i metodi matematici per la soluzione di problemi chimico-fisici servendosi del compilatore **Tubo Pascal**. Il prezzo irrisorio del volume (solo **17mila lire** per 176 pagine) può rappresentare un valido incentivo per gli interessati, studenti di scuola media o universitari.

richiesta del manuale del C/128 (ovviamente in italiano!) che sono disposto ad acquistare, anche in fotocopia, a qualsiasi prezzo.

Robero Ottani
Via Bologna 30
S. Giovanni in Persiceto (Bo)

Pubblico volentieri il disperato annuncio, anche se in netta contraddizione con la lettera precedente (impaginata apposta in posizione ravvicinata). La serietà di un'azienda, infatti, si vede anche da questi piccoli particolari. Un invito ai lettori che volessero accontentare il nostro Roberto: abbiate pietà, e non approfittate delle parole "a qualsiasi prezzo"...

Formattazione totale

Un programma - utility che posseggo consente di formattare un dischetto, inserito nel drive 1541, anche se è presente la tacchetta di protezione. Come è possibile una cosa del genere?

(Luciano Colombo - Milano)

Di solito un drive contiene un sensore direttamente collegato alla testina di lettura / scrittura che impedisce, se disattivato, di operare in scrittura anche se il software tenta di effettuare tale operazione. Il programma citato dal nostro lettore, probabilmente, riesce ad aggirare l'ostacolo e consente di realizzare la formattazione. Non riesco a capire, però, la reale utilità di una simile in-utility che, pericolosissima per i distratti (e non solo per loro), è potenzialmente in grado di commettere danni rilevanti.

Numeri complessi

Il mio C/64, alla richiesta di Print Sqr(-4), segnala un errore omettendo di visualizzare la notazione "immaginaria" 2i. E' normale il suo comportamento?

(Nicola Borelli)

Altrochè! Nessun linguaggio, almeno finora, implementa la possibilità di "trattare" numeri complessi. E' però possibile ricorrere ad un mini programma per elaborare anche questi casi particolari:

```
100 Input "Numero"; x: d=0
110 if X<0 Then d=1: x=-x
120 Print "Radice = ";
130 If d=1 Then Print "i";
140 Print Sqr(x): goto 100
```

Non appena il programma si accorge che il valore introdotto è negativo (riga 110), lo trasforma in positivo, ne elabora la radice quadrata e, prima di visualizzare il risultato, stampa il simbolo "i" (riga 130).

Sconcerto

Potete dirmi che cosa è il mouse? Mi piacerebbe collaborare con la vostra rivista dal momento che ritengo di essere piuttosto bravo a programmare...

(nome omesso per pietà)

La prima domanda fa nascere un leggerissimo sospetto sulla "bravura" vantata dal nostro lettore. Sarebbe come dire: "Vorrei partecipare al concorso per piloti di jet; a proposito: che cos'è un aereo?"...

Digitalizzare

Come si può realizzare la digitalizzazione di immagini televisive?

(Luigi Pulli - Massafra)

MS-DOS PROFESSIONAL 3
Mensile di software
L. 18.000
GESTIONE DI MAGAZZINO
IL PRIMO SOFTWARE GESTIONALE AL PREZZO DI UN LIBRO
Distr.: Parrini
Questo programma è proposto da
personal COMPUTER
di personal computer e workstation
Lire 6.000 La rivista per utenti
il grande mensile indipendente di software pratico per utenti di PC
IN EDICOLA
ANNO 1 - N. 3 - OTT 1990

Titoli della collana

CONTABILITA' GENERALE

•

GESTIONE FATTURAZIONE

•

GESTIONE DEL MAGAZZINO

•

CONTABILITA' IVA

Anche su dischetti da 3"5

Questi programmi sono disponibili anche in dischetti da 3,5" per i computer Olivetti Prodest PC1, Philips, ecc.

Per informazioni e richieste telefonare a: Systems Editoriale, via Mosè, 22 - 20090 Opera (MI) - Tel. (02) 55500310

BUONO SCONTO PER IL RACCOGLITORE AD ANELLI

Per ricevere l'apposito raccoglitore ad anelli a sole L. 7.000, comprese le spese di spedizione, invece di L. 10.000 più spese di spedizione, inviate questo tagliando a:
Systems Editoriale, via Mosè, 22 - 20090 Opera (MI)

EDIZIONI:
Systems Editoriale srl (Registro Nazionale Stampa n. 01500 vol. 15 foglio 793)

DISTRIBUZIONE:
Parrini
Pubblicazione registrata presso il Tribunale di Milano N. 215/81
Resp.: Michele di Pisa

systems

I segnali rintracciabili in un apparecchio TV, o in un videoregistratore, possono esser "processati" (=elaborati) da opportuni dispositivi che, collegati ad un computer, catturano le immagini grafiche e le registrano su supporto magnetico per poterle elaborare, a parte, con comodo.

La procedura, applicabile anche alle telecamere, è estremamente semplice ed alla portata di (quasi) tutte le tasche. Sia per il **C/64**, che per l'**Amiga**, sono disponibili digitalizzatori di vari prezzi, legati questi alla complessità della scheda elettronica adoperata.

Alcuni digitalizzatori, infatti, consentono la cattura dell'immagine solo se questa risulta perfettamente immobile per alcuni secondi, necessari al circuito elettronico per "scandire" l'intera schermata.

Ciò significa che sono possibili, con questo tipo di apparecchi, riprese con telecamere puntate su soggetti immobili (ritratti, fotografie), con videoregistratori dotati di fermo-immagine perfetti, con programmi televisivi che trasmettono... il monoscopio.

Circuiti più complessi, e quindi più costosi, consentono la digitalizzazione di immagini in tempo reale; ciò significa che l'elettronica utilizzata è in grado di scandire l'area video con estrema velocità, comunque inferiore al 25mo di secondo impiegato, dal sistema televisivo, per cambiare quadro.

Le immagini, ovviamente, vengono registrate su disco in un formato compatibile con i più diffusi pacchetti grafici utilizzabili con il computer adoperato.

Moltissime immagini che vedi pubblicate sulle nostre pagine non sono altro che digitalizzazioni, opportunamente modificate da abili video-disegnatori.

Espansioni C/64

Vorrei "espandere" il mio C/64 con un drive e con un mouse. Quale modelli mi consigliate?
(Luca Garulli)

Penso che il modello migliore sia l'A-500, meglio noto come Amiga 500. Se, però, invece di avventurarti in imprese anti - economiche, vuoi buttare i soldi dalla finestra, dimmi da quale finestra li butterai, in modo che qualcuno della Redazione possa recuperarli...

Domanda oscura

Sono una principiante e non riesco assolutamente a scrivere un programma in grado di svolgere le seguenti funzioni (segue elenco).
(Paola Mai - Avezzano)

Non mi è assolutamente chiara l'elaborazione richiesta.
Prova ad inviarmi l'ipotetica digitazione, da effettuare "sotto programma", che sarebbe necessaria per giungere alla soluzione.
Per esempio:
Scrivi n. di richieste **2**
Scrivi la funzione desiderata **somma**
Scrivi operaz. 1 **34**
Scrivi operaz. 2 **54**
Il risultato è 88
In *corsivo* è indicato il messaggio del computer mentre, in **neretto**, ciò che occorre digitare in fase di Input.

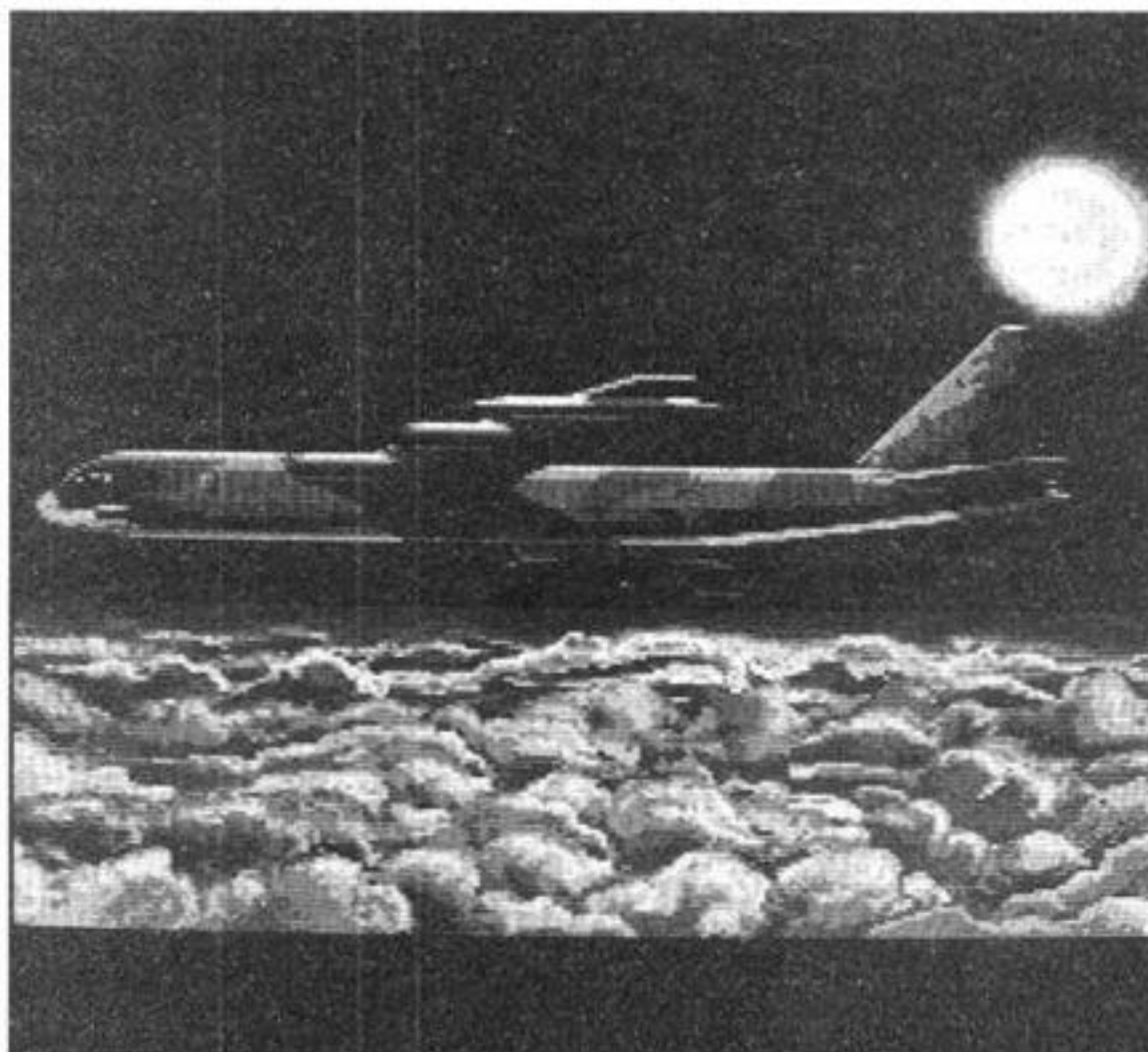
Trucchi sporchi

Quando contatto una BBS viene sempre richiesta la digitazione di nome, cognome, indirizzo e telefono.
Vorrei sapere se, invece di digitare i miei dati, digitassi il nome di un altro abbonato SIP, la conversazione viene a questi addebitata oppure no.
(nome omesso per vergogna)

Accà nisciuno è fesso. Perfino la Sip si è adeguata alla modernità, e la domanda non è tanto sciocca come sembra.

Una ventina di anni fa, infatti, componendo il numero della Sip relativo alla ricerca di abbonati, la centralinista di turno addebitava gli scatti (dovuti per il servizio reso) al numero comunicato da chi effettuava la telefonata, senza controllare se questi dicesse, o meno, la verità!

Oggi, per fortuna, gli scatti vengono addebitati automaticamente alla linea da cui parte la richiesta ed il pericolo di pagare scatti mai effettuati è scomparso.

CHI TROV
TROVA U

Quando si svolge un lavoro per "conto terzi", è assolutamente indispensabile che l'autore della consulenza comprenda perfettamente le esigenze del proprio cliente, anche se questi, paradossalmente, non sempre è in grado di esporle correttamente.

Ricordo, ad esempio, che alcuni anni fa, quando i data base "universali" dovevano ancora nascere, fui contattato da un'agenzia di ricerca del personale ed incaricato di scrivere un programma che fosse in grado di archiviare nominativi per rintracciarli, in seguito, in base ad una chiave.

Era, ricordo, l'epopea del **Pet Commodore** ed il cliente, dal momento che poteva avere forti sconti per questo computer, mi chiese di sviluppare il programma sulla linea di prodotti Commodore.

Fin dall'inizio compresi che il cliente non aveva le idee chiare e fui costretto ad ingaggiare una lotta furibonda per fargli acquistare il Top della linea Pet (modello **8032** dotato di ben 32K Ram, doppio drive da 1.2 megabyte ciascuno) inducendolo a rinunciare alla (sua) primitiva idea di limitarsi al **Pet 2001** dotato del solo registratore a cassette...

E feci bene!

Quando, infatti, presentai una prima bozza del programma (che elaborava, inutile dirlo, files relativi) il cliente richiese di passare dai 2000 nominativi, concordati all'inizio, a circa 10000(!) per essere al riparo da eventuali, futuri incrementi.

Alla seconda stesura, che teneva conto delle nuove richieste, il cliente si accorse che avrebbe preferito la ricerca in base a più campi contemporanei (chiave di lettura **And** e **Or**). Al terzo appuntamento il cliente richiese la stampa su carta delle ricerche, nonostante egli stesso avesse preteso (per evitar cartacce) che la ricerca venisse effettuata esclusivamente "a video". Una volta riscosso l'assegno, consegnato il programma (opportunamente "protetto") ed istruito per un'intera giornata la segretaria (mooolto carina. Ah! se avessi dieci anni di meno...) fui contattato per l'ennesima volta dal mio cliente perchè il suo Direttore, entusiasta del programma che aveva avuto occasione di veder funzionare, non riusciva ad effettuare una copia dello stesso per uso personale...

Nonostante io sia un sostenitore convinto dell'uso dei prodotti biodegradabili, non lo invitai a fertilizzare autonomamente il suo giardino, ma solo perchè sono molto educato.

La morale della favola, quindi, invita ad esser previdenti e, soprattutto, a mettersi nei panni di chi, realmente, utilizzerà il computer.

Riferendomi all'esempio di prima, durante i primi contatti (ahimè, solo professionali) con la segretaria, mi resi conto che era assolutamente indispensabile inserire una routine di Input controllato che, nei programmi che di solito scrivo a mio personale uso e consumo, non mi sogno nemmeno di prendere in considerazione.

Ciò perchè, nella digitazione, alla ragazza capitava di premere il tasto sbagliato, di inserire un carattere minuscolo invece di uno maiuscolo (o viceversa), di premere anzitempo un tasto di selezione, e così via.

Il programma, pertanto, privo di una provvidenziale routine di controllo, prendeva direzioni spesso indesiderate, con grande sconforto della ragazza che non sapeva, giustamente, che cosa fare.

Le speranze economiche di vari lettori offrono lo spunto per esprimere alcuni suggerimenti su come "far soldi" con il computer

Le regole d'oro

Quando si scrive un programma per conto terzi, pertanto, è assolutamente indispensabile seguire i suggerimenti che leggerete tra breve: eviterete di litigare e, nella migliore delle ipotesi, di perdere tempo.

1 - L'utente. Nel prendere accordi, pretendete di conoscere la persona (o le persone) che avranno l'incarico di utilizzare il computer. Non sempre si tratta di ragazze carine, nè di addetti particolarmente propensi a sobbarcarsi la fatica di imparare qualcosa di nuovo. A volte, incredibile a dirsi, ci si trova di fronte ad individui che, soprattutto se digiuni di informatica, considerano un elaboratore con ostilità e non perderanno occasione di manifestare il loro dissenso dichiarando di non capirci nulla, o entrando per errore(?) nel menu sbagliato o evitando accuratamente i controlli preliminari necessari: verifica di collegamento con la stampante o corretto inserimento del foglio di carta, chiusura dello sportellino del drive, mancato inserimento del dischetto, corretta successione di digitazioni varie.

Se avete diplomazia, è il momento di tirarla fuori; se non l'avete, cercate di lavorare in tandem con qualche vostro conoscente (che presenterete come "collega", anche se totalmente ignorante in informatica), che sia in grado di appianare eventuali difficoltà di comprensione.

Un "collega" ben vestito, giacca e cravatta sono d'obbligo, è assolutamente indispensabile soprattutto se siete molto giovani ed altrettanto esperti; a pochi

A UN AMICO N TESORO

titolari d'azienda fa piacere parlare d'affari con individui in giovane età, soprattutto se sulla vostra faccia è perfettamente leggibile un'espressione del tipo "Sta' zitto, vecchio caprone, che non capisci un piffero di computer".

Cercate, quindi, di comprendere a fondo la predisposizione d'animo dell'utente reale (l'operatore, insomma, e non il Direttore che vi pagherà); se questi dimostra interesse, voglia di apprendere e di migliorare la propria posizione, siete a cavallo: avrete un alleato che, se contattato dal Cliente per sapere come procedono le cose, parlerà a vostro favore; sempre che il programma che state scrivendo sia davvero valido...

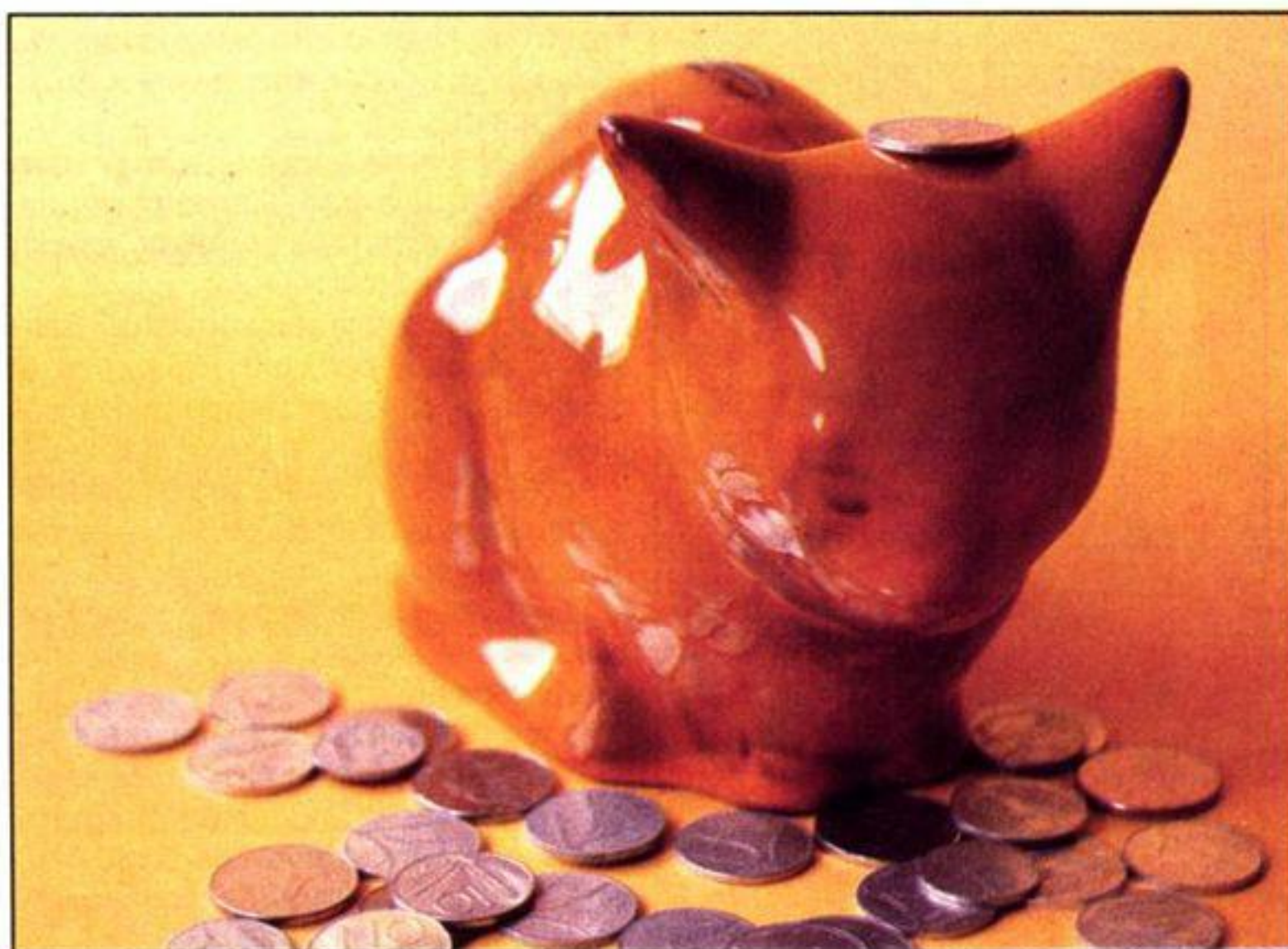
2 - Il programma. A meno che non siate dei veri maghi, in grado di sfruttare abilmente tutte le risorse del linguaggio preferito e del sistema operativo, non imbarcatevi, almeno all'inizio della vostra professione, nella proposta di programmi ad hoc. Procuratevi un ottimo spreadsheet (**Quattro**, della Borland, va più che bene), un diffuso **word processor** (WordStar è uno dei più blasonati), un valido **Data Base** (a vostra scelta, ce n'è per tutti i gusti); tutti in versione originale, per via del manuale di istruzioni, rigorosamente indispensabile per capirci qualcosa. Imparate ad usarli a menadito e sfruttateli al massimo delle loro potenzialità: vi accorgerete che, opportunamente programmati, saranno in grado di accontentare numerosissime esigenze di carattere generale. Contemporaneamente vi renderete conto dei limiti offerti da ciascun programma e non rischierete di fare brutte figure proponendo una procedura non realizzabile, almeno in tempi brevi.

Di solito, infatti, è molto più semplice trovare una richiesta di consulenza per la prima installazione di computer in un ufficio (formazione del personale, insomma) che non la richiesta di un programma specifico. Ma anche in questo caso, comunque, vi accorgerete che uno spreadsheet o un data base ben programmati potranno accontentare il vostro cliente. In seguito, e sempre se il cliente sarà soddisfatto del vostro modo di operare, questi tornerà da voi se avrà bisogno di nuove consulenze; nel frattempo, se siete abbastanza furbi, imparerete ad usare programmi professionali sempre più specifici (tra cui **fatturazione, contabilità, I.V.A.**) che avrete acquistato, in versione originale, con una parte del ricavato della vostra consulenza; e così via verso competenze, tecniche ed... economiche, di sempre maggior rilievo.

3 - Il computer. Se il computer che dovrete usare esiste già, nel senso che il Cliente lo possiede, prendete nota delle sue caratteristiche (tipo di sistema, capacità del disco rigido, n. di drive e loro formato, versione DOS e così via) e, se possibile, portatelo a casa per lavoraci "sopra"; la compatibilità con l'elaboratore in vostro possesso potrebbe non esser completa e rischiereste di svolgere un lavoro inutile. Non fatevi prendere dalla tentazione, comunque, di sfruttare particolarità della scheda grafica ivi montata: cambiando computer, il programma sviluppato con tanta fatica potrebbe non funzionare come vi aspettate!

Rifiutatevi di lavorare su computer vecchi, come i primitivi sistemi **CP/M**, C/64, C/128, anche se in configurazione "da parata", per un semplice motivo: se il potenziale cliente non è in grado di acquistare, per lavoro, un computer realmente professionale, è molto probabile che debba limitare drasticamente anche la cifra necessaria per la vostra consulenza.

Se, invece, il cliente ritiene necessario cambiar computer, o acquistarne uno più al passo con i tempi, non lasciatevi prendere dalla tentazione di farci sopra la cresta, magari mettendovi d'accordo con un negoziante: vi squalifichereste di colpo, senza rimedio, per una manciata di biglietti da diecimila. Anzi, chiedete al cliente di segnalarvi un fornitore di sua fiducia e fate in modo da fargli risparmiare, per quanto possibile, sull'hardware. Ciò significa che potete evitare di far acquistare uno schermo a colori, e relativa scheda grafica, se il lavoro da svol-

gere sarà prevalentemente sulla Triade (Word processor / Desk Top Publishing, Data Base, Spreadsheet): una scheda Hercules in bianco e nero è molto più indicata per un uso intensivo dell'elaboratore; accontentatevi di una memoria da 1 megabyte (pretendete l'espansione di più mega nel caso di DTP, si opera più velocemente): pochi programmi necessitano di una memoria di maggiori dimensioni che, in moltissimi casi, non verrebbe sfruttata appieno; per ciò che riguarda i drive, non scendete al di sotto di 40 mega (per l'hard disk) e fate montare un drive da 3.5 ed uno da 5.25, non necessariamente ad alta capacità, per quanto, ormai, i prezzi non siano molto distanti tra loro.

Il coprocessore matematico lasciatelo perdere a meno che, ovviamente, non dobbiate impostare un programma commerciale che lo "riconosca" e ne faccia buon uso.

Nel risparmiare, però, non esagerate: se vi chiedono di installare un computer per usarlo esclusivamente come word processor, un sistema basato su un 8088 / 8086 potrebbe andare più che bene. Ma **un elaboratore AT, basato su di un 80286, ormai poco più caro di un XT**, sarà utile nel caso vi vengano affidate, in un prossimo futuro, consulenze più impegnative; l'accettazione della proposta di un AT vi consentirà di scoprire, inoltre, se il potenziale cliente tende, o meno, al... risparmio. Un computer 80386, ovviamente, impedirà l'avanzata precoce dell'obsolescenza e vi permetterà, in seguito, di proporre pacchetti applicativi che su un semplice 80286 non potrebbero girare.

Il **mouse** deve esserci, e deve essere di ottima qualità.

Per la stampante non sparate subito l'acquisto di una laser: non sapete ancora come sceglierla, altrimenti non leggereste questo articolo(!).

Scartate immediatamente le stampanti a sfera o a margherita: nel caso di riproduzione di grafici vi trovereste in difficoltà; una stampante a 9 aghi vi permetterà di tenere basso il vostro preventivo e potrà essere sempre utilizzata come stampante "secondaria", quando deciderete che è giunto il momento di far acquistare anche una laser.

4 - Il coraggio. Se siete alla prima esperienza, non perdetevi d'animo. Un po' di sfrontatezza è preferibile alla remissività; ricordate sempre che il vostro potenziale Cliente non capisce un tubo di computers oppure non ha il tempo di occuparsene; altrimenti non chiederebbe aiuto: ha bisogno di qualcuno che possa risolvere i suoi problemi; perchè non dovreste esser proprio voi? Se siete capitati al momento giusto, quindi, datevi da fare e fate valere le vostre competenze.

Cercate, magari, di mettervi d'accordo con alcuni rivenditori di computers. Questi, di solito, non hanno assolutamente il tempo di offrire altri servizi se non la vendita. L'offerta di installazione, quindi, potrebbe essere un'occasione valida per aumentare il "prestigio" del punto vendita (per il negoziante) e di realizzare qualche liretta extra (per voi).

Non chiedete mai una cifra inferiore alle cinquantamila lire per la prima installazione di un computer: con questo intendo il puro e semplice collegamento di monitor, computer, stampante; la formattazione dell'hard disk, la creazione di una subdirectory in cui sistemare tutti i file del Dos (e quelli consigliati dal fabbricante) la stesura di un mini file-batch in cui, oltre alla comparsa del vostro nome cognome indirizzo e telefono, venga caricato automaticamente il file che riconosce la tastiera ed, eventualmente, la scheda grafica adoperata. Non dimenticate di "regalare" un mini word processor di pubblico dominio (che voi conoscete benissimo, ma che sia privo di istruzioni o, possibilmente, con istruzioni in tedesco) ed approfittatene per verificare il corretto funzionamento della stampante.

Se il Cliente ha anche acquistato un pacchetto applicativo, installatelo, e non dimenticate di proporre un vostro minicorso di quattro / sei ore a non meno di 30mila l'ora.

Se, infine, avete giochini simpatici e programmi vari di intrattenimento, regalateli con noncuranza (non più di un dischetto, però) ma sottolineate con vigore il fatto che il regalo è un extra dovuto alla vostra innata generosità: avrete tutto da guadagarci.

Infine, regalate anche un **virus- scan** e mettete in guardia il Cliente (che ormai sarà anche diventato un ottimo amico) dai pericoli di infezioni.

Cercate, prima di salutarlo (e di incassare il denaro, per carità) di capire di quali future consulenze può aver bisogno.

Accennate con serietà, ma con discrezione, alle vostre tariffe orarie: eviterete di farvi scroccare, gratis, informazioni "volanti", magari telefoniche, anche se queste sono limitate ai programmi che avrete regalato.

La competenza, non dimenticatelo mai, **deve** essere remunerata; se regalate troppo, anche se per pura gentilezza o calcolata "promozione", la gente penserà che non valete un granchè; e vi tratterà di conseguenza...

CAMPUS

64/128

SOMMARIO

Nessun albero viene abbattuto per gli inserti di Commodore Computer Club, stampati su carta riciclata al 100%

UN JOY DA... ACCAREZZARE

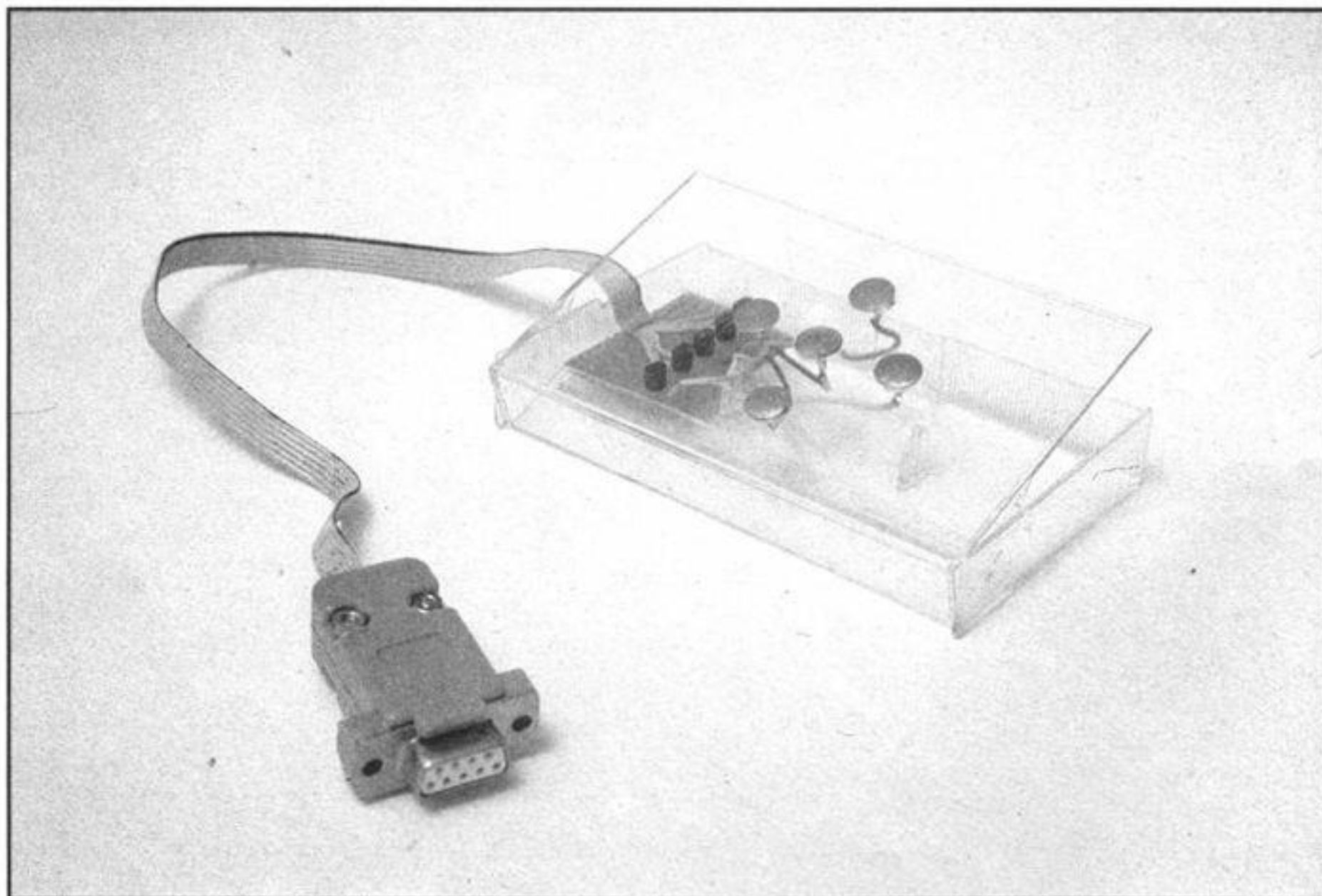
E' sufficiente sfiorare i "bottoni" del semplice apparecchio per simulare i movimenti di un comune joy elettromeccanico. I vantaggi? Elevata velocità di movimento; ma non è tutto oro ciò che luccica

di Giuseppe Bisanti

Il principale difetto di un joystick è costituito dalla presenza di elementi meccanici

Il joystick che presentiamo è del tipo **sensor**, applicazione diffusissima nel campo dell'elettronica, hobbistica etc. Praticamente si avvale del contatto epidermico (niente di erotico), per azionare un componente, nel nostro caso un transistor. Non si può certo pretendere la supervelocità, ma le prestazioni sono abbastanza soddisfacenti, tenuto conto della estrema semplicità circuitale. Si possono realizzare altri joysticks, basati su circuiti simili, ricorrendo a soluzioni elettroniche diverse: uso di CMOS, aggiunta di altri transistor per aumentare l'affidabilità. Il risultato potrebbe portare ad un "aggeggio" forse più rapido, ma non particolarmente gradito a chi col saldatore ha poca dimestichezza; si aggiunga, poi, la necessità di inserire componeti passivi (condensatori, resistenze). Ciò che presentiamo, ad ogni buon conto, rappresenta di certo una base di partenza per soluzioni più ardite.

Lo schema elettrico

Lo schema elettrico si riferisce a cinque **darlington NPN tipo BC517**, componente pra-

AVVERTENZE

L'apparecchio descritto in queste pagine deve esser realizzato esclusivamente da chi è realmente in grado di effettuare connessioni elettriche ed elettroniche (in particolare, digitali) con la massima competenza possibile; in ogni caso è consigliabile la verifica del funzionamento prima di effettuare collegamenti con computer o suoi accessori.
La Systems Editoriale e l'autore del progetto pubblicato, pertanto, declinano ogni responsabilità da danni che dovessero eventualmente verificarsi, anche a causa di errori di stampa, di impaginazione e di progettazione hardware o software.

ticamente identico ad un transistor "vulgaris", con la differenza che all'interno vi sono **due** transistors (che **non** pagherete al prezzo di uno..).

Si possono utilizzare anche dei normali transistors tipo NPN o PNP (recuperandoli, magari, da apparati in disuso), collegando, in questo caso, il collettore al posto dell'emettitore.

Il darlington relativo ad una funzione del joystick è collegato al corrispondente piedino della porta controller: pin 1, 2, 3, 4 (per le quattro direzioni); 6 (pulsante di sparo); massa, collegata al pin 8.

Affinchè una funzione del joystick venga attivata, basta modificare uno dei primi cinque Bit del registro corrispondente alla porta, praticamente, resettando (azzerando il bit, brutalmente cortocircuitando a massa), il pin in questione. E' questo il compito svolto dai pulsanti contenuti in un normale joystick elettromeccanico.

Nel nostro caso, invece, il darlington si comporta come un **pulsante elettronico**, che viene azionato non da una pressione, bensì dal contatto del dito(ne).

Osservando il semplice schema, noterete che la base del darlington è connessa con un sensore che ha la funzione di "captare" impulsi parassiti dovuti alla corrente alternata.

Quando il sensore non è sfiorato, il darlington non conduce; al contrario, grazie alla piccola tensione presente sulla base, si commuta nello stato di conduzione allorchè si producono parassiti elettrici, toccando il sensore.

Nello schema, tra collettore e massa di T5, è inserito un condensatore elettrolitico da 1uF (tratteggiato) che, se inserito, migliora (in parte) gli spostamenti, rendendoli più morbidi.

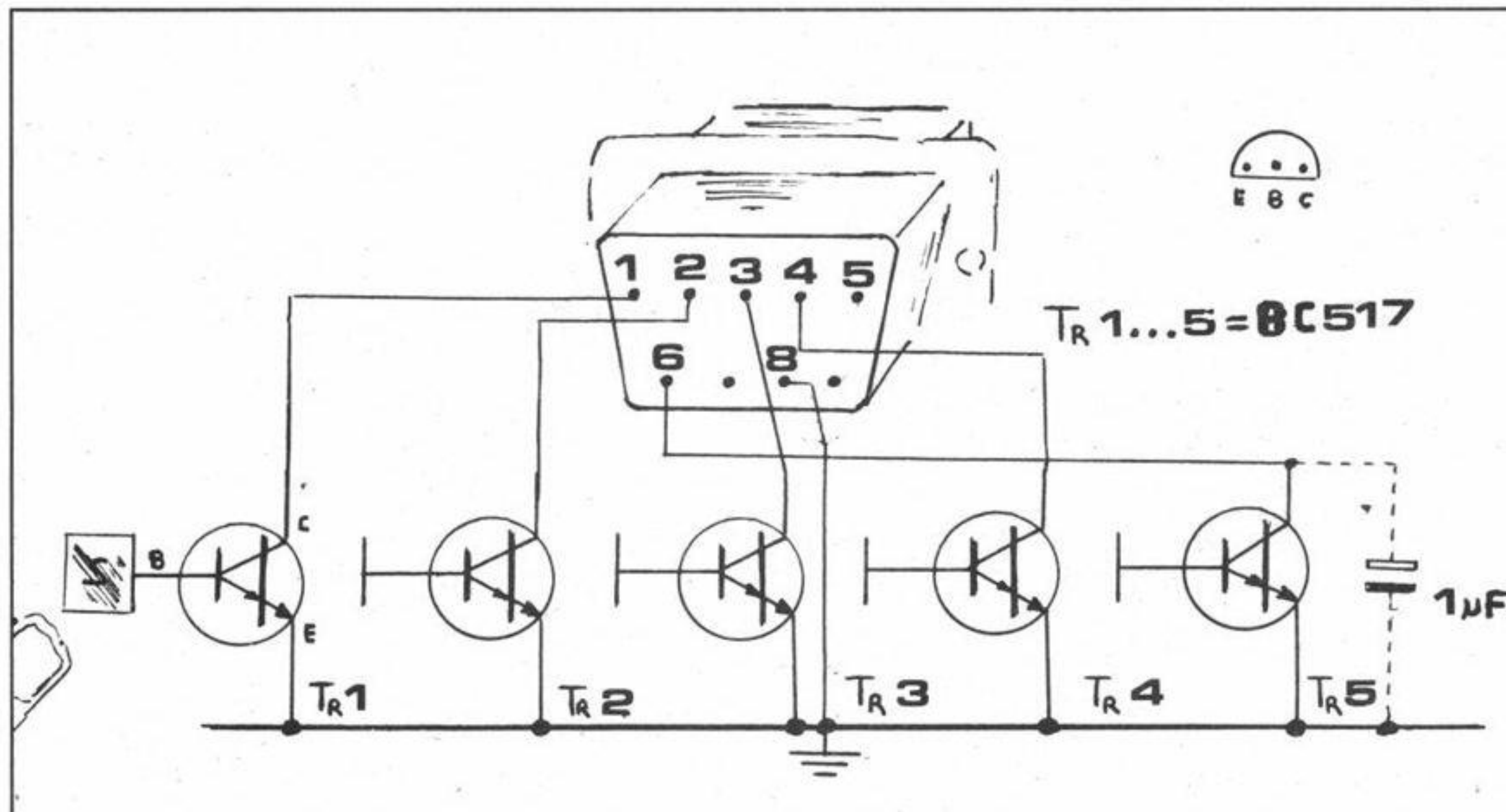
Ovviamente di condensatori occorrerà montarne altri quattro: uno per ogni darlington, nella stessa configurazione.

Con quest'ultima variante, però, sarà necessario collegare il joystick **dopo** il caricamento del programma che ne richiede l'utilizzo.

L'assorbimento (corrente di riposo) dei transistors è di pochi milliampere e non dovrebbe pregiudicare il funzionamento del computer.

Alla porta controller, infatti, si possono prelevare max 50 mA (caso del C/64).

Si potrebbe, inoltre, inserire una resistenza di circa 330 Kohm (tra base e massa) per aumentare la sensibilità; si potrebbero, addirittura, in-

Il progetto è molto semplice, alla portata dei principianti

I più esperti potranno apportare migliorie circuitali

serire dei led (che ovviamente non servono a nulla, ma fanno effetto..), il tutto a discapito della semplicità.

Almeno nella realizzazione del prototipo, quindi, conviene attenersi alla configurazione pubblicata, con la quale i risultati sono accettabili.

I "transistors" sono montati su una basetta forata per circuiti sperimentali, sostituibile con un qualsiasi supporto isolante (anche il cartone). Per quanto concerne i **sensori**, sono state scelte delle normali **puntine da disegno** che, preventivamente stagnate, non rischiano di ossidarsi.

Come sensori si potrebbero usare placchette di metallo, reperibili nelle cassette magnetiche in disuso. La scelta è vasta, e lo è anche per la custodia; nelle foto di queste pagine il circuito risulta inserito nel contenitore plastico di una comune musicassetta. Qualunque sarà il tipo di sensore che adotterete (purchè metallico), dovrete forare la plastica del contenitore.

A tale scopo sarà sufficiente appoggiare sullo stesso la punta metallica di un chiodo e riscaldarla con il saldatore. Attraverso i fori passeranno i fili da saldare sulle puntine, che poi fisserete con mastice adesivo; la superficie dei sensori va pulita con alcool dai residui delle saldature.

In alcuni casi i sensori sono tanto sensibili in particolari circostanze (vicinanza di lampade, conduttori, resistenza del corpo, etc.) che basta avvicinare il dito (senza toccarli), per abilitarli.

In altri casi, invece, sarete costretti ad umettarvi le dita affinchè si creino gli impulsi "parassiti".

In ogni caso conviene lasciare, tra i sensori, una certa distanza (1 / 2 cm.) in modo che la "presenza" del dito non sia avvertita contemporaneamente da due elementi.

Si possono realizzare i sensori incidendoli su di una basetta di bakelite per circuiti stampati, dandogli la forma che vi aggrada.

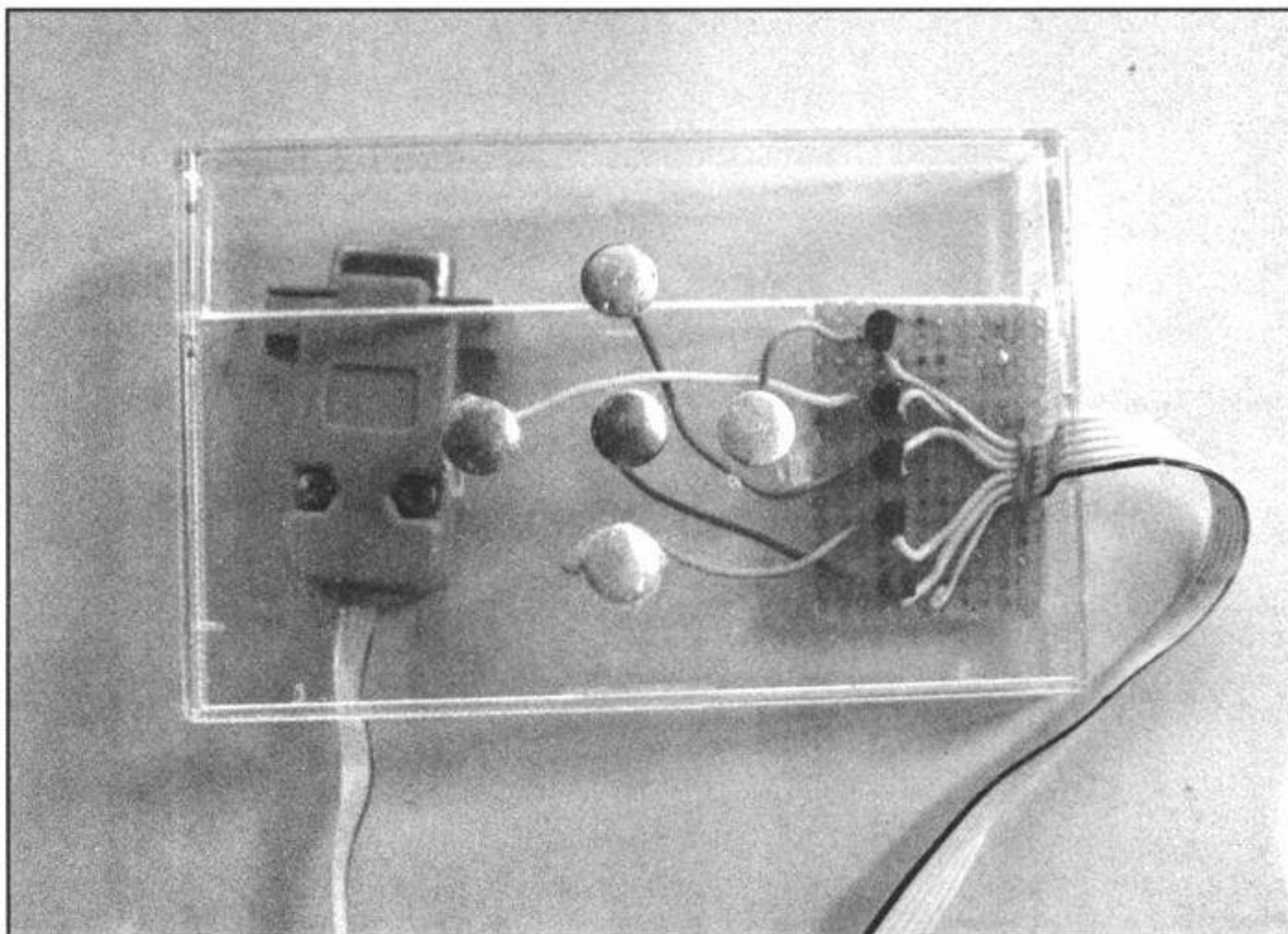
I sensori si possono fissare prima, saldando i fili dopo, a patto di usare un saldatore di bassa potenza.

Per collegare la basetta al connettore, si consiglia di usare piattina a sei conduttori, preferendola al cavo, in quanto più flessibile, pratica e reperibile.

La corrispondenza fra i contatti del connettore giochi, e le funzioni dei sensori, è riportata nello schema elettrico (1= alto; 2= basso; 3 = sinistra; 4 = destra; 6 = fuoco; 8 = massa).

Il funzionamento dello joy è garantito, ma subordinato al giocatore, "captatore" di parassiti dovuti alla corrente alternata, e quindi alle sue condizioni fisiologiche (resistenza suerficiale della pelle).

Può risultare un valido test per la parassitosi...

Se proprio ci tenete, personalizzate il joy con le vostre impronte digitali, premendo fortemente, con l'indice, la puntina bagnata di stagno fuso, ben caldo...

MARE MOSSO SULLO SCHERMO!

E' sufficiente manipolare i vettori di Interrupt per realizzare simpatici effetti, utili nella presentazione di vostri programmi

di Filippo Bruno

Chi traffica spesso con dischetti copiati avrà sicuramente osservato, magari a bocca aperta, gli innumerevoli effetti spettacolari con i quali i pirati condiscono le loro presentazioni.

Uno tra i più appariscenti è sicuramente il **movimento ondulatorio** di una o più righe di schermo (un effetto simile si trovava anche nella presentazione di un dischetto della **Systems**).

Il programma l.m. proposto in queste pagine genera proprio questo effetto, pur se limitatamente alla prima riga dello schermo.

Esso è da considerarsi puramente esemplificativo della tecnica adottata, in quanto la gestione di più righe (facile dopo che si è capito il meccanismo) comporterebbe l'aggiunta di una maggiore mole di dati da inserire nelle tabelle usate.

Ai più smanettoni, come al solito, il compito di modificare a piacimento la routine.

Vecchie conoscenze

Il programma riprende abbondantemente quello di **Lorenzo Emilitri** del numero 58 di C.C.C. (articolo "**Tutti i tempi del video**") e manipola la tabella dei ritardi riguardo alle linee che vede il VIC.

Il programma, che gira in interrupt, è composto da due routines distinte, corrispondenti ad altrettanti scatti video da parte del raster.

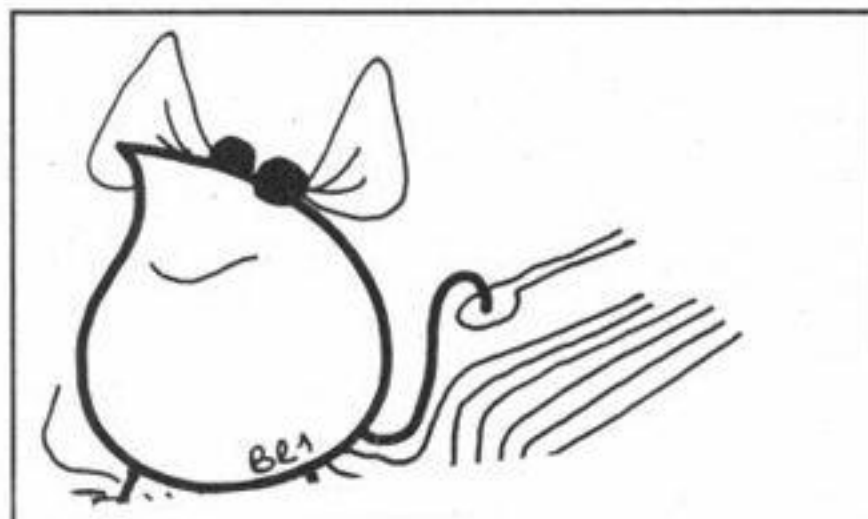
Infatti il programma non fa altro che mettere gli otto dati (uno per ciascuna riga del cursore) nel registro **$D016** di scroll orizzontale; in seguito genera uno scatto nella riga **250** dello schermo, che, essendo l'ultima, non interferisce sul corretto funzionamento dello schermo.

Quando viene richiamato l'interrupt, se il sistema si accorge che lo scatto verificatosi è quello in basso allo schermo, trasferisce 8 dati dall'area valori (lunga 208 dati) in quella (lunga 8) che viene letta nella prima routine; subito dopo viene ripristinato lo scatto alla riga superiore.

Il byte chiamato **Switch** (nel disassemblato di queste pagine), ossia interruttore, è utilizzato per rallentare la velocità delle onde, in quanto costringe a scrivere per due volte consecutive gli stessi 8 dati; sostituendo, infatti, la locazione **$F8** del controllo prima dell'istruzione Assembly **BNE**, posta all'inizio della seconda routine, con un altro byte **non vitale** per il sistema operativo (come ad esempio **$F7**), noterete un raddoppio della velocità.

Inoltre, confrontando la tabella dei tempi usata nel programma, noterete che il primo valore è 1 e non 6.

Questo perchè il programma richiede tempo (cioè cicli macchina) per controllare se la richiesta di IRQ deve esser considerata valida per il primo oppure per il secondo scatto.

In verità il byte di ritardo dovrebbe valere **0.5**, ma non essendo possibile (operiamo in "semplice" linguaggio macchina!) dovremo accontentarci di qualche leggero sfarfallio nella parte destra del video, comunque accettabile.

Ancòra una manipolazione dell'Interrupt; digitare per credere!

```
100 rem ********************************
110 rem * ondulafont by f.bruno soft ! *
120 rem *                              *
130 rem *                              *
140 rem * (c) 1990          06839578   *
150 rem ********************************
160 :
170 fori=49152 to 49283:rem $c000->$c083 incluso
180 read a:c=c+a:poke i,a:next
190 if c<> 19518 then print"▓errore nei data!":stop
200 c=0:for i=49408 to 49616:rem $c000->$c0d0
210 read a:c=c+a:poke i,a:next
220 if c<> 42328 then print"▓errore nei data dell'area valori!":stop
230 c=0:for i=49617 to 49624:rem $c1d1->$c0d8
240 read a:c=c+a:poke i,a:next
250 if c<> 57 then print"▓errore nei data dell'area tempi!":stop
260 print"▓per attivare digitare : sys 49152"
270 end
280 rem ********************************
290 rem *per salvare la routine diretta*
300 rem *mente in $c000-$c183 e poterla*
310 rem *cosi' caricare istantaneamente*
320 rem *con load'ondulafont-49152',8,1*
330 rem *digitare  :  run 370          *
340 rem *nb:viene anche salvata a parte*
350 rem *la tabella con i valori usati.*
360 rem ********************************
370 print"▓":print"inserire un disco con almeno 3 blocchi"
380 print:print"liberi. poi premere un tasto."
390 poke198,0:wait198,1:poke198,0
400 f$="ondulafont":l=0:h=192:x1=49152:x2=49283:gosub440
410 f$="ondulafont.tab":l=0:h=193:x1=49408:x2=49624:gosub440
420 print:print"fatto.":end
430 rem-----------------------------
440 open15,8,15,"i0"
450 open8,8,8,f$+",p,w"
460 print#8,chr$(l);chr$(h);
470 fori=x1 to x2
480 read a:print#8,chr$(a);:next:rem attenzione al punto e virgola!
490 close8:input#15,a,a$,b,c:print:printa,a$,b,c:print#15,"i0":close15:return
500 rem-----------------------------
510 data120,169,45,162,192,141,20,3,142,21,3,162,0,142,14,220
520 data134,249,232,142,26,208,169,250,141,18,208,173,17,208,41,127
530 data141,17,208,169,200,162,193,133,251,134,252,88,96,169,1,141
540 data25,208,173,18,208,201,224,176,40,162,0,189,249,192,188,209
550 data193,136,208,253,141,22,208,234,234,232,224,8,208,237,169,200
560 data141,22,208,169,250,141,18,208,165,249,73,255,133,249,76,188
570 data254,165,249,208,23,24,165,251,105,8,201,208,208,2,169,0
580 data133,251,160,8,177,251,153,248,192,136,208,248,169,50,141,18
590 data208,76,49,234
600 rem-----------------------------
610 rem- dati dell'area valori scroll-
620 rem-----------------------------
630 data0,201,200,200,200,200,200,200,200,202,201,200,200,200,200,200
640 data200,203,202,201,200,200,200,200,200,204,203,202,201,200,200,200
650 data200,205,204,203,202,201,200,200,200,206,205,204,203,202,201,200
```

```
660 data200,207,206,205,204,203,202,201,200,207,207,206,205,204,203,202
670 data201,207,207,207,206,205,204,203,202,207,207,207,207,206,205,204
680 data203,207,207,207,207,207,206,205,204,207,207,207,207,207,207,206
690 data205,207,207,207,207,207,207,207,206,206,207,207,207,207,207,207
700 data207,205,206,207,207,207,207,207,207,204,205,206,207,207,207,207
710 data207,203,204,205,206,207,207,207,207,202,203,204,205,206,207,207
720 data207,201,202,203,204,205,206,207,207,200,201,202,203,204,205,206
730 data207,200,200,201,202,203,204,205,206,200,200,200,201,202,203,204
740 data205,200,200,200,200,201,202,203,204,200,200,200,200,200,201,202
750 data203,200,200,200,200,200,200,201,202,200,200,200,200,200,200,200,201
760 rem--------------------------------
770 rem- dati dei tempi di ritardo   -
780 rem--------------------------------
790 data1,8,8,8,8,8,8,8
800 end

ready.
```

```
;********************************
;* caratteri  ondulatori  sulla *
;* prima  riga   dello  schermo *
;********************************
areava =$c100
tempi  =$c1d1
scroll =$d016
switch =$f9
exit   =$ea31
valori =$c0f8
       *=$c000
;-------------------------------
       sei                ;dirotta l'irq
       lda #<start        ; a start
       ldx #>start
       sta $0314
       stx $0315
       ldx #$00
       stx $dc0e          ;reset timer
       stx switch         ;azzera
       inx                ;'interruttore'
       stx $d01a          ;set irqs
       lda #$fa           ;scatto all'ultima
       sta $d012          ;riga visibile
       lda $d011          ;reset msb raster
       and #$7f
       sta $d011
       lda #<areava+200;prepara i
       ldx #>areava       ;puntatori
       sta $fb            ;per il trasferimento
       stx $fc            ;dei dati scroll
       cli
       rts                ;esce
;-------------------------------
start  lda #$01           ;conferma irq
       sta $d019
       lda $d012          ;e' nella parte
       cmp #$e0           ;bassa?
       bcs trasl          ;si-> salta
       ldx #$00           ;legge 8 valori
loop   lda valori+1,x     ;da mettere nel
       ldy tempi,x        ;registro di scroll
loop1  dey                ;e aspetta un
       bne loop1          ;proporzionale
       sta scroll         ;ai cicli richiesti
       nop                ;per evitare
       nop                ;sfarfallii
       inx
       cpx #$08
       bne loop
       lda #$c8           ;rimette valore
       sta scroll         ;standard di scroll
       lda #$fa           ;nuovo scatto
       sta $d012          ;in basso
       lda switch         ;scambia il valore
       eor #$ff           ;del registro
       sta switch         ;interruttore
       jmp $febc          ;esce
;-------------------------------
trasl  lda switch         ;se e' 1 aspetta
       bne esce           ;un'altra chiamata
       clc                ;aggiunge 8 al
       lda $fb            ;puntatore
       adc #$08           ;se questo supera
       cmp #$d0           ;208 lo ripristina
       bne dopo           ;riazzerandolo
       lda #$00           ;trasferisce dalla
dopo   sta $fb            ;area valori 8
       ldy #$08           ;dati mettendoli
loop2  lda ($fb),y        ;da $c0f9 a $c100
       sta valori,y
       dey
       bne loop2
esce   lda #$32           ;rimette scatto alto
       sta $d012
       jmp exit           ;esce
       .end
```

A CACCIA DI SET E DI SPRITE

Due programmi, rigorosamente in linguaggio macchina, utili per rintracciare i set di caratteri ridefiniti e gli sprite dei videogames preferiti; e ricopiarli a volontà...

di Filippo Bruno

Una procedura in l.m. per esplorare la memoria del C/64-128

Qualsiasi gioco serio fa uso di caratteri ridefiniti che vengono utilizzati sia per cambiare la grafia dei caratteri da utilizzare nei messaggi (spesso si ricorre a due o più caratteri per formarne uno solo, in modo da renderlo più grande e appariscente), sia per disegnare i fondali, ossia la scenografia di un gioco.

Spesso il fondale è in movimento e l'utilizzo di un font specifico semplifica le operazioni che il programmatore deve compiere; si pensi alla necessità di scrollare una bitmap a colori...

Seguendo il procedimento descritto in queste pagine sarete in grado di sproteggere (e, magari, salvare a parte) qualsiasi set di caratteri ridefinito da qualsiasi gioco, a patto di non confondere una bitmap o un blocco di sprites per un set e purchè possediate il tasto di reset.

VIETATO AI MINORI

E' probabile che i neo-utenti di computer non riescano a ben comprendere l'utilità della procedura software descritta nelle presenti pagine. Questa è infatti destinata a coloro che sono già padroni delle principali tecniche di programmazione e desiderano approfondire il modo in cui un elaboratore organizza i vari dati all'interno della memoria.
Il programma, pertanto, rappresenta un invito ai lettori più esperti, soprattutto a coloro che, considerando il listato come una base di partenza, riescano a pervenire a procedure più interessanti e capaci, magari, di offrire applicazioni di più ampio respiro.
Se hai incominciato da poco, comunque, non scoraggiarti!
Tutti coloro che, oggi, vantano una particolare competenza nel campo dell'informatica hanno iniziato con un banale Print "Pippo".
Perchè non dovresti riuscire anche tu?...

Allocando

Trascrivete il primo listato in Basic (**Show character set**) ponendo la massima attenzione alle linee contenenti i **Data**.

Il programma sarà anche in grado di salvare, su disco, i valori del codice macchina contenuti nell'area compresa tra 832 e 904 ($0340, $0388) con il nome di **Show+trasf.-832**.

E' ora utile rinfrescarsi le idee su come il circuito VIC del C/64 gestisce i set di caratteri. Come tante altre cose (bitmap, sprites...) non possiamo allocare fonts dove ci pare, ma abbiamo a disposizione zone già predisposte, che possono essere così localizzate da uno specifico puntatore (53272 - $D018).

Ad esempio, le zone possibili (oltre al banco di default, che il VIC "vede" all'accensione) possono essere quelle indicate in tabella.

Non tutte le zone, però, possono essere selezionate a piacere. Infatti da $0000 a $07FF c'è la cosiddetta pagina zero, contenente una miriade di puntatori vitali, l'area di schermo ed altri dati importanti, per non parlare del fatto che, dopo un reset, una buona parte dell'area stessa viene ripulita; ecco perchè il programma presente a partire dalla locazione 832 non interferirà con i dati del set da rintracciare.

Se, poi, si seleziona la zona da $1000 a $17FF oppure da $1800 a $1FFF, il chip preleva i caratteri non dalla Ram, bensì dai dati dei due set di default (maiuscolo e minuscolo) posti in Rom da $D000 a $DFFF.

Sono rimaste da esplorare, quindi, l'area da $0800 a $0FFF (che si visualizza con **Poke 53272, 18**) e quelle da $2000 a $3FFF, che si visualizzano inserendo, rispettivamente nel re-

gistro 53272 i valori riportati in tabella tramite una banale **poke 53272, x** (con x = 24, 26, 28 oppure 30).

Dopo aver caricato il gioco desiderato, (e, ovviamente, resettato), caricate il programma **Show+trasf.-832** con la sintassi **,8,1** che avrete registrato in precedenza, digitando **Run 320** (vedi **Rem** da riga 260 a 300).

Esso consiste in due programmi, il primo dei quali equivale al Basic...

For i=0 to 255: poke 1024 + i, i: next

...che visualizza l'intero set di caratteri "attuale" e parte con **Sys 832**.

Dopo averlo lanciato, cambiate il contenuto di 53272.

Se individuate il set di caratteri cercato, ricordate da quale locazione parte e in quale termina (tale valore deve essere aumentato di 1) e inserite, nel formato high byte / low byte, i due indirizzi nelle locazioni 43 / 44 e 45 / 46.

Subito dopo (**non** digitate List o altri comandi, pena un possibile inchiodamento!) eseguite un normalissimo...

Save "nome del font", n, 1

...con **n = 8** per salvarlo su disco oppure **n = 1** per salvarlo su nastro.

```
100 rem **********************************
110 rem *    show   character   set    *
120 rem *    +  trasferisce  memoria    *
130 rem *   (c) filippo bruno soft      *
140 rem **********************************
150 :
160 fori=832 to 904:rem $0340->$0388 incluso
170 read a:c=c+a:poke i,a:next
180 if c<> 10339 then print"█errore nei data!":stop
190 print"█":print"  per visualizzare il set digitare :"
200 print:print"  sys 832"
210 print:print:print"  per trasferire la memoria da"
220 print:print"  $x000 a $(x+2)000 in $2000/$3fff"
230 print:print"  digitare : sys 842,x"
240 end
250 rem **********************************
260 rem *per salvare la routine diretta*
270 rem *mente in $0340-$0388 e poterla*
280 rem *cosi' caricare istantaneamente*
290 rem *con load'show+trasf.-832',8,1 *
300 rem *digitare  :  run 320          *
310 rem **********************************
320 print"█":print"inserire un disco con almeno 1 blocco"
330 print:print"libero. poi premere un tasto."
340 poke198,0:wait198,1:poke198,0
350 open15,8,15,"i0"
360 open8,8,8,"show+trasf.-832,p,w"
370 print#8,chr$(64);chr$(3);
380 fori=832 to 904
390 read a:print#8,chr$(a);:next:rem attenzione al punto e virgola!
400 close8:input#15,a,a$,b,c:print:printa,a$,b,c:print#15,"i0":close15
410 print:print"fatto.":end
420 rem--------------------------------
430 data162,0,138,157,0,4,232,208,249,96,32,253,174,32,158,183
440 data32,130,3,133,252,232,232,32,130,3,133,250,160,0,132,251
450 data120,169,52,133,1,132,248,169,32,133,249,177,251,145,248,200
460 data208,249,230,252,230,249,165,252,197,250,208,239,169,55,133,1
470 data88,96,138,24,42,42,42,42,96
480 end
```

```
;-------------------------------
;- show character set + trasf. -
;- da $x000 a $(x+2)000 in $2000-
;- (c) filippo bruno soft - roma-
;-------------------------------
        *=$0340
        ldx #$00        ;riempie lo schermo
loop    txa
        sta $0400,x     ;con i codici in x
        inx             ;da 0 a 255
        bne loop
        rts             ;esce
        jsr $aefd       ;controllo virgola
        jsr $b79e       ;parametro in x
        jsr rotate      ;lsb in msb
        sta $fc         ;lo mette nel
        inx             ;puntatore
        inx
        jsr rotate      ;msb=msb+2
        sta $fa         ;lo mette nel
        ldy #$00        ;puntatore di controllo
        sty $fb
        sei
        lda #$34        ;set ram sotto rom
        sta $01
        sty $f8         ;prepara puntatori
        lda #$20        ;destinazione
        sta $f9
loop1   lda ($fb),y     ;esegue il
        sta ($f8),y     ;trasferimento
        iny
        bne loop1
        inc $fc         ;incrementa la
        inc $f9         ;parte bassa
        lda $fc         ;e se e' = a $fa
        cmp $fa         ;deduce che ha
        bne loop1       ;finito.
        lda #$37        ;ripristina rom su ram
        sta $01
        cli
        rts             ;esce
rotate  txa             ;acc=x
        rol             ;msb=lsb
        rol
        rol
        rol
        rts             ;torna
        .end
```

Digitate con attenzione le linee Data, pena il blocco del computer

Se, ad esempio, avete rintracciato il set nell'area da $2800 a $2FFF (controllate con **Poke 53272, 26** che sia quello giusto) per salvarlo occorre che digitiate:

Poke 43,0: Poke 44,40: Poke 45,0: Poke 46,8

Da notare che i valori presenti dopo la virgola devono essere in **decimale** ed occorre, quindi, trasformare gli indirizzi eventualmente rintracciati in notazione **esadeciamale**. Infatti a $28 corrisponde 40; $2F (+1) = $30 = 48. Quando, poi, aggiungete 1 a $2FFF (cioè $3000) il valore decimale da inserire nella locazione 46 sarà 0 e non 255 (=$FF).

Per farla breve, l'indirizzo finale (già incrementato di uno) corrisponde all'indirizzo iniziale del successivo set di caratteri.

E se non viene visualizzato nessun set significativo?

Allora vuol dire che i dati si trovano in un altro banco di memoria (ve ne sono 4). Per non esser costretti a spostare font, cambiare banco, memoria video, puntatori sprites, eccetera, il secondo programma in lm, che parte da 842 ($034A), provvede a spostare una qualsiasi zona lunga 8192 bytes (tra quelle possibili) nell'area da $2000 a $3FFF.

Per trasferire, ad esempio, l'area compresa tra $4000 e $5FFF in $2000 / $3FFF, basta digitare **Sys 842, 4**, in cui 4 è il primo numero dell'indirizzo iniziale.

Il programma provvede anche a manipolare la locazione 1 per "vedere" le Ram poste sotto la Rom e sotto l'interprete basic, proprio per "scovare" qualsiasi font.

In seguito potrete salvare il set come descritto prima. Per evitare numerosi tentativi potete subito, al momento del reset, stabilire se il set è nel banco di default o se occorre fare dei trasferimenti.

Se, infatti, quando lo schermo si restringe appaiono caratteri sensa senso, ma con la grafia del Basic, allora il set è nel banco di default, altrimenti no.

Ciò avviene perchè, al momento del restringimento del video, la CPU ripristina il valore 21 in 53272 ma "dimentica" di ripristinare subito il banco di default!

Suggerimenti utili: per selezionare il modo multicolor digitare **Poke 53270, 216** ed inserire i due registri colore supplementari in **53282** e **53283**. Con colori del cursore tra 0 e 7 si ottengono caratteri standard; con i colori tra 8 e 15 si hanno caratteri in multicolor. Se x = 24 si può selezionare x = 26 semplicemente premendo, contemporaneamente, i tasti Shift e Commodore, praticamente passando al minuscolo; stessa cosa per passare da x = 28 a x = 30.

Se vi accorgete che un'immagine o un testo, visualizzati con **Sys 832**, si interrompe a metà schermo (ossia in corrispondenza del carattere 255 di un set, con x = z) e riprende con x = z + 2, allora quella è una bitmap, perchè il raster non può effettuare una interruzione a metà schermo orizzontalmente.

Eccezioni

Non è tutto Basic ciò che sembra: nel gioco **Maniac Mansion**, infatti, il set dei testi non è il set standard minuscolo, ma è uno simile, abilmente manipolato; al carattere "**a**", ad esempio, corrisponde "**p**"; ad "**s**" la virgola "**,**" eccetera.

Il programmatore del gioco, quindi, è ricorso a tale strattagemma per impedire l'individuazio-

```
100 rem **********************************
110 rem * show sprites allocati tra     *
120 rem *         $2000 & $3fff         *
140 rem *   by  filippo bruno soft      *
150 rem **********************************
160 :
170 fori=49152 to 49539:rem $c000->$c183 incluso
180 read a:c=c+a:poke i,a:next
190 if c<> 49664 then print"♥errore nei data!":stop
200 print"♥":print"per attivare digitare : sys 49152"
210 print:print:print:print" f1 - scorre avanti gli sprites"
220 print:print" f3 - scorre indietro gli sprites"
230 print:print" f5 - accende o spegne il multicolor"
240 print:print" f7 - torna in basic"
250 end
260 rem **********************************
270 rem *per salvare la routine diretta*
280 rem *mente in $c000-$c183 e poterla*
290 rem *cosi' caricare istantaneamente*
300 rem *con load'show mobs-49152',8,1 *
310 rem *digitare   :  run 330          *
320 rem **********************************
330 print"♥":print"inserire un disco con almeno 2 blocchi"
340 print:print"liberi. poi premere un tasto."
350 poke198,0:wait198,1:poke198,0
360 open15,8,15,"i0"
370 open8,8,8,"show mobs-49152,p,w"
380 print#8,chr$(0);chr$(192);
390 fori=49152 to 49539
400 read a:print#8,chr$(a);:next:rem attenzione al punto e virgola!
410 close8:input#15,a,a$,b,c:print:printa,a$,b,c:print#15,"i0":close15
420 print:print"fatto.":end
430 rem----------------------------------
440 data169,147,32,210,255,162,8,169,15,157,38,208,202,208,250,162,17
450 data189,93,193,157,255,207,202,208,247,142,32,208,142,33,208,142
460 data16,208,142,28,208,169,2,141,37,208,169,5,141,38,208,169
470 data255,141,23,208,141,29,208,141,21,208,169,128,133,249,162,0
480 data169,36,133,250,32,67,193,162,2,32,67,193,162,4,169,48
490 data133,250,32,67,193,162,6,32,67,193,169,48,141,109,5,141
500 data253,6,169,52,141,118,5,141,6,7,169,56,141,127,5,141
510 data15,7,169,3,141,136,5,141,24,7,165,197,201,6,208,11
520 data173,28,208,73,255,141,28,208,76,215,192,201,3,208,21,169
530 data0,141,21,208,169,6,162,14,141,33,208,142,32,208,169,147
540 data32,210,255,96,201,4,208,17,24,165,249,105,8,133,249,208
550 data4,169,128,133,249,72,76,228,192,201,5,208,19,56,165,249
560 data233,8,133,249,201,127,16,4,169,248,133,249,72,76,228,192
570 data165,249,72,76,228,192,160,58,162,255,202,208,253,136,208,248
580 data76,123,192,162,0,165,249,157,248,7,232,224,8,240,5,230
590 data249,76,230,192,169,50,133,250,104,133,249,234,234,234,234,74
600 data74,72,41,240,201,32,240,2,230,250,162,8,32,67,193,162
610 data10,32,67,193,162,12,104,41,15,201,9,16,8,24,105,48
620 data133,250,76,54,193,160,1,217,126,193,240,4,200,76,40,193
630 data132,250,76,54,193,32,67,193,162,14,230,250,32,67,193,76
640 data215,192,189,110,193,133,251,189,111,193,133,252,162,4,169,0
650 data168,165,250,145,251,152,24,105,9,202,208,244,96,36,79,107
660 data79,180,79,250,79,36,159,107,159,180,159,250,159,106,5,250
670 data6,254,6,110,5,107,5,251,6,108,5,252,6,0,10,0,12,0,14
680 end

ready.
```

```
;********************************
;*  visualizza  gli sprites da  *
;*       $2000 a $3fff          *
;* (c) filippo bruno       1990 *
;********************************
;
primo   =$f9
vic     =$d000
video   =$fa
        *=$c000
;------------------------------
        lda #$93        ; print
        jsr $ffd2       ; chr$(147)
        ldx #$08        ;
        lda #$0f        ; colore=15
loop    sta vic+38,x    ;
        dex             ;
        bne loop        ;
        ldx #$11        ;
loop1   lda coord-1,x   ; legge le
        sta vic-1,x     ; coordinate
        dex             ; dalla tabella
        bne loop1       ;
        stx $d020       ; bordo e sfondo
        stx $d021       ; neri
        stx vic+16      ; x>255 register
        stx vic+28      ; e mult. spenti
        lda #$02        ; colori per
        sta vic+37      ; multicolor
        lda #$05        ;
        sta vic+38      ;
        lda #$ff        ; accende tutti
        sta vic+23      ; i mobs e li
        sta vic+29      ; espande
        sta vic+21      ;
        lda #$80        ; primo valore=128
        sta primo       ; ( =>$2000 )
        ldx #$00        ; dispone i $
        lda #$24        ; ( $24=$ )
        sta video       ;
        jsr dollar      ;
        ldx #$02        ;
        jsr dollar      ;
        ldx #$04        ; dispone gli 0
        lda #$30        ; ( $30=0 )
        sta video       ;
        jsr dollar      ;
        ldx #$06        ;
        jsr dollar      ;
        lda #$30        ; dispone i
        sta $056d       ; 0,4,8,c
        sta $06fd       ;
        lda #$34        ;
        sta $0576       ;
        sta $0706       ;
        lda #$38        ;
        sta $057f       ;
        sta $070f       ;
        lda #$03        ;
        sta $0588       ;
        sta $0718       ;
;------------------------------
;-grande  loop per  il controllo-
;-della  pressione  dei tasti f.-
;------------------------------
loooop  lda $c5         ; legge 197
        cmp #$06        ; e' f5 ?
        bne dopo2       ; no->salta
        lda vic+28      ; si->inverte
        eor #$ff        ; bit del
        sta vic+28      ; multicolor
        jmp pausa       ; esce
dopo2   cmp #$03        ; e' f7 ?
        bne dopo3       ; no->salta
        lda #$00        ; spegne mobs
        sta vic+21      ;
        lda #$06        ; colori originali
        ldx #$0e        ; del bordo
        sta vic+33      ; e dello sfondo
        stx vic+32      ;
        lda #$93        ; print
        jsr $ffd2       ; chr$(147)
        rts             ; esce in basic
dopo3   cmp #$04        ; e' f1 ?
        bne dopo4       ; no->salta
        clc             ; si->aggiunge
        lda primo       ; 8 al valore
        adc #$08        ; del primo
        sta primo       ; puntatore
        bne oltre1      ; se e' 0 allora
        lda #$80        ; e' 128
        sta primo       ;
oltre1  pha             ; salva il valore
        jmp write       ; aggiorna
dopo4   cmp #$05        ; e' f3 ?
        bne ripeti      ; no->salta
        sec             ; si->sottrae
        lda primo       ; 8 al valore
        sbc #$08        ; del primo
        sta primo       ; puntatore
        cmp #$7f        ; se e' <127
        bpl oltre2      ; e' 248
        lda #$f8        ;
        sta primo       ;
oltre2  pha             ; salva
        jmp write       ; e aggiorna
ripeti  lda primo       ;
        pha             ; salva ugualmente
        jmp write       ; e aggiorna
;------------------------------
pausa   ldy #$3a        ; pausa per
loop9   ldx #$ff        ; distinguere
loop8   dex             ; bene i mobs
        bne loop8       ; (il lm e'
        dey             ; troppo veloce!)
        bne loop9       ;
        jmp loooop      ; chiude il ciclo
;------------------------------
write   ldx #$00        ; aggiorna i
loop3   lda primo       ; vettori
        sta $07f8,x     ; (2040->2047)
        inx             ;
        cpx #$08        ;
        beq oltre       ;
        inc primo       ;
        jmp loop3       ;
oltre   lda #$32        ; aggiorna i
        sta video       ; primi 2 bytes
        pla             ; dopo il $
        sta primo       ; controllando
        lsr             ; prima il msb
        lsr             ; del primo
        pha             ; valore, poi
        and #$f0        ; il lsb
        cmp #$20        ;
        beq dopo        ;
        inc video       ;
dopo    ldx #$08        ;
        jsr dollar      ;
        ldx #$0a        ;
        jsr dollar      ;
        ldx #$0c        ;
        pla             ;
        and #$0f        ;
        cmp #$09        ; se lsb e' <10
        bpl abcdef      ; vi aggiuge
        clc             ; 48 (vedi tabella
        adc #$30        ; codici video
        sta video       ; sul manuale)
        jmp write1      ;
abcdef  ldy #$01        ; se lsb e' una lettera
loop4   cmp numeri,y    ; lo trova
        beq dopo1       ; e ottiene
        iny             ; il codice video
        jmp loop4       ; in y
dopo1   sty video       ;
        jmp write1      ;
;------------------------------
```

```
write1 jsr dollar          ; scrive il primo            lda #$00        ; per 4 volte
       ldx #$0e            ; valore sopra        loop2  tay             ; il codice video
       inc video           ; e il primo +1              lda video       ; ( $fa )
       jsr dollar          ; sotto                      sta ($fb),y     ;
       jmp pausa           ; esce                       tya             ;
;---------------------------------                      clc             ;
dollar lda tabell,x        ; legge le coordinate        adc #$09        ;
       sta $fb             ; dello schermo              dex             ;
       lda tabell+1,x      ; dove depositare            bne loop2       ;
       sta $fc             ; a intervalli               rts             ;
       ldx #$04            ; di 9 spazi

;--------------------------------
coord  .byte 36,79,107,79,180,79,250,79,36,159,107,159,180,159,250,159
tabell .byte $6a,5,$fa,6,$fe,6,$6e,5,$6b,5,$fb,6,$6c,5,$fc,6
numeri .byte 0,10,0,12,0,14
       .end
```

ne dei testi analizzando il disco, settore per settore.

Basterà trasferire, con **Sys 842, 14**, la zona \$E000 / \$FFFF in \$2000 / \$3FFF, salvarla (è lunga 9 blocchi) nell'area riservata ai caratteri minuscoli. In seguito bisognerà utilizzare un disk monitor con il set cifrato (occorre sapere i comandi del monitor a memoria!) per leggere, in maniera molto più comprensibile, tutti i testi presenti nel gioco.

Inizio	Fine	X
\$0000	\$07FF	16
\$0800	\$0FFF	18
\$1000	\$17FF	21
\$1800	\$1FFF	23
\$2000	\$27FF	24
\$2800	\$2FFF	26
\$3000	\$37FF	28
\$3800	\$3FFF	30

Tabella
Valori da digitare nella locazione 53272 per attivare i vari set di caratteri.

A caccia di mobs

La tecnica utilizzata per impadronirsi di set di caratteri ridefiniti può essere adottata anche per la ricerca di sprites e per il loro salvataggio.

Per ottenere ciò copiate il secondo programma Basic (**Show sprites**) facendo sempre molta attenzione alle linee contenenti le istruzioni **Data**.

Il listato carica nella memoria (e vi consigliamo di salvare anche il l.m. puro su disco con **Run 330**) a partire da **49152** (\$C000) un programma in l.m. che permette di esplorare 128 sprites per volta, non a caso proprio quelli allocati da \$2000 a \$4000 (escluso), così da utilizzare la routine di trasferimento (**Sys 842, x**) utilizzata precedentemente.

E' anche possibile selezionare il modo multicolor.

Purtroppo la sua lunghezza, superiore a 192 bytes, non ha consentito di allocarlo nell'area compresa tra \$0340 a \$0400.

Per esplorare tutte le zone possibili, quindi, occorrerà, dopo aver resettato il gioco che interessa:

1) caricare il programma "Show+trasf.-832"; caricare il programma "Show mobs-49152", lanciarlo con Sys 49152 ed esplorare la zona;

2) trasferire in 842 la zona da \$0000 a \$1FFF in \$2000 / \$3FFF ed esplorarla;

3) trasferire a mano a mano le altre aree di 8192 bytes in \$2000 tranne quella da \$C000 a \$DFFF, perchè in parte contiene il programma esploratore;

4) ricaricare il gioco da cui si vogliono estrarre gli sprites, resettare, caricare il programma in 832 e trasferire **subito** la zona \$C000 / \$DFFF con **Sys 842, 12**; quindi caricare il programma in \$C000 ed esplorare la zona.

Per salvare i dati dei mobs la tecnica è identica a quella utilizzata per salvare i dati dei set ridefiniti, alterando cioè i puntatori di inizio e fine basic 43 / 44 e 45 / 46 con l'inizio e la fine (più uno) dell'area interessata.

E' bene sottolinare che il **Macroassembler**, incontrando il codice **LSR** (che effettua la divisione per due traslando i bit di un posto a destra), non riconosce tale istruzione, benchè valida per l'Assembly "ufficiale" del 6510; possono quindi verificarsi sorprese nell'esaminare il codice oggetto generato dal Macroassembler.

Alcuni programmi Assemblatori possono presentare problemi in fase di compilazione

AUTOCAD, SCRIVILO IN BASIC

Il C/64 dotato di emulatore Gw-Basic, oppure un computer Ms- Dos compatibile, possono "supportare" un mini-autocad per la rappresentazione di figure geometriche

di Giancarlo Mariani

Il listato che proponiamo è un altro esempio di come si possano far funzionare programmi identici sia sul **C/64 dotato di emulatore Gw-Basic** che sul più potente linguaggio originale degli elaboratori Ms-Dos.

"**Microcad**" (così è denominato il listato di queste pagine) funziona infatti indistintamente sulle due macchine; a patto che, naturalmente, entrambe siano dotate dell'interprete **Gw- Basic**.Come avrete già capito dal nome, MicroCad è un programma che serve per disegnare dato che emula perfettamente, in ogni particolare, la versione 10.0 di Autocad, che normalmente gira su Ms-Dos 386 con 4 MB di memoria e coprocessore matematico... (magari).

Scherzi a parte, MicroCad si propone, come d'altronde tutti i listati di CCC, di essere un programma didattico; di cose da imparare, anche guardandolo superficialmente, ve ne sono una miriade. Potrete, ad esempio, dare uno sguardo alla gestione dei files sequenziali, alla grafica, alla gestione di tabelle, ma soprattutto potrete imparare a conoscere meglio il buon vecchio Gw-Basic che, pur girando anche sull'ormai modesto C/64, permette di avvicinarsi al mondo Ms-Dos.

Come gira Microcad

Descriveremo, ora, le funzioni del programma e, in seguito, il funzionamento del listato. MicroCad, una volta digitato e fatto partire, propone il seguente Menu:

1) Carica disegno
2) Salva disegno
3) Vedi/Modifica composizione disegno
4) Inserisci composizione disegno
5) Vedi grafica disegno
6) Cancella disegno
7) Fine MicroCad

Ovviamente, ammesso che sia la prima volta che facciamo "girare" MicroCad, dovrà essere scelta l'opzione 4, ossia l'inserimento del disegno. Premendo il tasto 4, pertanto, MicroCad propone un nuovo menu:

1) Linee
2) Rettangoli
3) Cerchi
4) Ritorno al menu principale

Inutile dire che dovremo scegliere tra linee, rettangoli oppure cerchi, tramite i tasti **1**, **2** o **3** seguiti da Return. Se scegliamo linee o rettangoli verranno chiesti, in fase di input, quattro valori: X1, Y1, X2, Y2.

Nel caso siano state selezionate le **linee**, le variabili **X1** e **Y1** rappresenteranno le coordinate di partenza, mentre **X2**, **Y2** quelle di "arrivo" della linea stessa.

Per i **rettangoli**, invece, X1 e Y1 sono le coordinate dell'angolo in alto a sinistra, mentre X2 ed Y2 sono quelle dell'angolo in basso a destra.

Nel caso dei **cerchi** (o meglio, **circonferenze**), i numeri chiesti in input saranno sempre 4, ma questa volta chiamati **X**, **Y** (coordinate del **centro**), **R** (**raggio**) e **Aspetto**, che indica il "rapporto" che desideriamo tra X e Y; ricordiamo, infatti, che nella pagina grafica il numero dei punti in orizzontale non è uguale a quello in verticale.

L'accorgimento può servire per produrre artificiosamente cerchi "schiacciati" o "allungati" (quindi **ellissi**); su un video del formato 320 x 200 il rapporto deve valere circa 5/6 per produrre cerchi (quasi) perfetti.

Microcad chiederà i quattro valori per la prima linea (rettangolo o cerchio), quindi per la seconda, per la terza e così via fino ad **un massimo di 50 "elementi"** grafici.

I valori numerici vanno inseriti separati da virgole (ad esempio 10, 20, 30, 40) e, per terminare l'inserimento, sarà sufficiente digitare ***,,,** (asterisco seguito da 3 virgole) e premere Return.

Digitando "***,,,**" si ritornerà al menu di scelta tra linee, rettangoli e cerchi e quindi si potrà decidere se continuare ad inserire altri elementi oppure tornare al menu principale.

Copialo per telefono

Anche i listati presenti in queste pagine possono esser tirati giù per mezzo del modem; se, ovviamente, ne possedete uno.
La procedura per collegarsi con la nostra banca dati (attiva 24 ore su 24) è riportata su altra parte della rivista.

Il "vero" Autocad

Autocad è un potentissimo programma per effettuare disegni su elaboratori Ms-Dos compatibili, usato professionalmente da progettisti, ingegneri e disegnatori tecnici. Il programma possiede un'infinità di funzioni, tramite le quali è possibile fare su di un disegno praticamente ciò che si vuole: rotazioni 3D, zoom, sovrapposizioni, ingrandimenti, riduzioni e tante altre funzioni (c'è chi giura di averlo visto riprodurre dei falsi di Picasso...).

AutoCad consente di trattare i disegni in modo realmente **professionale**: quando registra un disegno su disco, **non** memorizza il disegno così come appare su video (cioè la schermata grafica), ma le istruzioni per comporlo, vale a dire l'intera sequenza di istruzioni tipo "Linee", "Cerchi", "Punti", e tutte le altre che servono per la riproduzione del disegno.

In pratica ciò che viene registrato da AutoCad è un vero e proprio programma che, al momento della rilettura, viene interpretato ed eseguito dal **Cad** (**Com**puter **Aided Design** = Disegno supportato da computer).

Quelli che hanno visto in azione AutoCad originale si saranno senz'altro resi conto che quando viene caricato un disegno, lo si vede proprio composto linea per linea, punto per punto (a meno che non si stia usando un Ms-Dos 386 con 33 Mhz di clock...)

Tenendo conto di questa caratteristica è stato sviluppato **MicroCad** che, come detto prima, è un programma puramente didattico, ma si presta ad ampliamenti e modifiche a non finire; chissà che un giorno un lettore di CCC non diventi famoso per aver sviluppato la versione 15.0 di AutoCad partendo dal listato di queste pagine...

MicroCad permette di disegnare linee, rettangoli e cerchi, quindi di memorizzare le istruzioni in una specie di database, da modificare a piacimento, da registrare o ricaricare da disco.

Continuando l'inserimento, questo continuerà a partire dall'elemento successivo all'ultimo inserito.

Le matrici di Microcad

Il programma si basa sull'elaborazione di tre matrici :

Li (1 .. 50, 0 .. 3) che contiene i dati relativi alle **linee**.

Re (1 .. 50, 0 .. 3) che si riferisce ai **rettangoli**.

Ce (1 .. 50, 0 .. 3) che contiene i **cerchi**.

In pratica: Li (X, 0) conterrà la coordinata X1 del segmento X-esimo, Li (X, 1) conterrà Y1, Li (X, 2) conterrà X2 mentre Li (X, 3) conterrà Y2 (il valore **X** indica di quale linea stiamo parlando). Questo vale anche per i rettangoli, cambiando ovviamente il nome dell'array in Re. Per i cerchi, Ce (X, 0) contiene X, Ce (X, 1) contiene Y, Ce (X, 2) contiene R e Ce (X, 3) contiene l'aspetto della circonferenza X-esima.

Per ognuna delle matrici esiste un **puntatore** (**Pl** per le linee, **Pr** per i rettangoli e **Pc** per i cerchi) relativo al numero di elementi inseriti nell'array sino a quel momento. La **routine di input** non fa altro che richiedere i quattro numeri componenti l'elemento del disegno (X1, Y1, X2, Y2 per linee e rettangoli, X, Y, R, Asp per i cerchi) e quindi inserirli negli array prima citati.

In effetti è necessario un altro array di "appoggio", chiamato **Te**, anch'esso dimensionato a 50 x 3.

Questo per evitare di scrivere tre routines distinte per l'input delle linee, dei rettangoli e dei cerchi.

I numeri a mano a mano digitati, infatti, verranno sempre inseriti nell'array temporaneo; in seguito, a seconda del tipo di figura prescelta, Te verrà travasato in Li, Re oppure Ce.

MicroCad non esegue il controllo dei valori digitati, che quindi andranno mantenuti entro i limiti dall'utente (**0 .. 319** per le coordinate X e **0 .. 199** per le Y).

Inoltre, non viene fatto alcun controllo sul numero degli elementi inseriti, e se si inseriscono più di 50 elementi, il programma si blocca fornendo un messaggio di errore.

Per **disegnare un punto** sarà sufficiente selezionare, dal menu, "Linee" e quindi inserire la coordinata X2 uguale a X1 e Y2 uguale a Y1.

Esempio di punto al centro dello schermo:

160, 100, 160, 100.

Una volta completata la nostra opera d'arte, converrà salvarla su disco sfruttando l'opzione 2 del menu principale. Richiamando "salva disegno", MicroCad proporrà il menu di scelta:

0) Tutto
1) Linee
2) Rettangoli
3) Cerchi
4) Ritorno al menu principale

Scegliendo 0 si salva l'intero disegno, mentre una scelta compresa tra 1 e 3 consente di salvare solo una parte di esso, appunto come indicato. Prima del salvataggio viene chiesto il **nome del disegno** da salvare, che verrà salvato in tre files differenti: "**nome.lin**" per le linee, "**nome.ret**" per i rettangoli e "**nome.cer**" per i cerchi. A fine salvataggio, MicroCad ritorna al menu precedentemente proposto.

Registrare i disegni

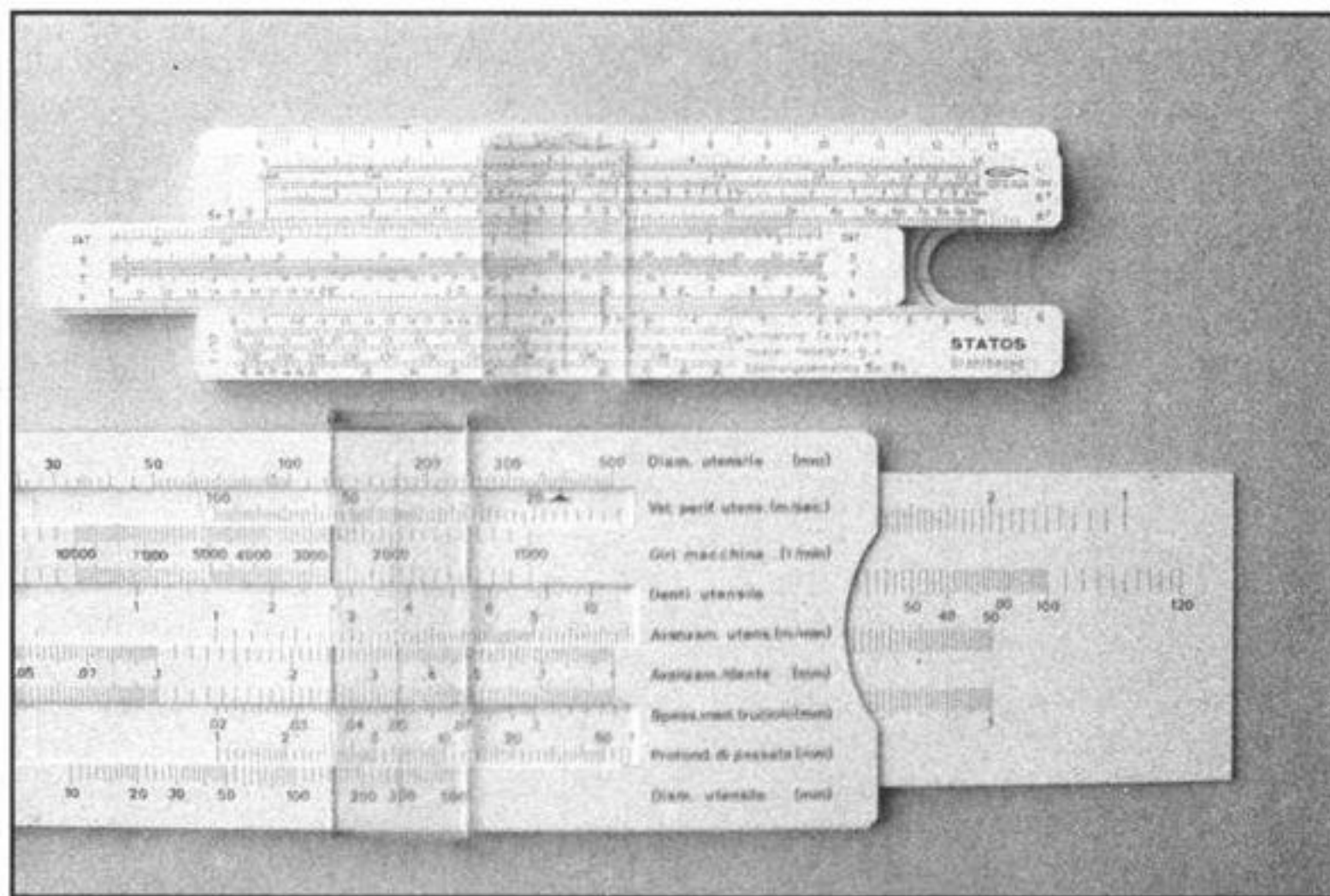
Il disegno viene salvato tramite semplici files sequenziali che, come si vede dal listato, richiedono la semplice elaborazione delle istruzioni **Open, Print#, Close**. In ciascuna linea del file vengono registrati, separati da virgole, i valori che compongono l'elemento del disegno.

Nella prima riga del file viene registrata un'**intestazione** che contiene, oltre al tipo di elemento registrato in quel file, anche il nome del disegno stesso. Questi files sono "ASCII standard" e possono quindi essere tranquillamente "visti" tramite il comando "Type" del dos (ed anche dell'emulatore Ms-Dos del C/64).

Ne consegue che è possibile creare figure servendosi di un banale **word processor** (o di un **editor**, altrettanto banale) in grado di registare files in formato Ascii "puro".

La procedura di salvataggio è semplicissima: viene eseguito un ciclo For... Next da 1 al numero di elementi contenuti in ciascuna matrice; il valore terminale, lo ricordiamo, è presente nelle variabili Pl, Pr, e Pc; vengono quindi salvati i quattro valori, che compongono l'elemento, su ogni riga del file.

MicroCad non esegue controlli di errore su disco; si suppone che, prima di eseguire la manovra di salvataggio, nel drive sia stato inserito un dischetto formattato e con spazio libero sufficiente per contenere i vari files. Inoltre bisogna stare attenti nell'assegnare il nome al disegno. Questo deve essere composto unicamente da caratteri accettati dal DOS o dal C/64 come nomi dei files. Se si inserisce il nome di un disegno già presente su disco, quest'ultimo verrà brutalmente cancellato, e quindi sovrascritto dal nuovo disegno.

A questo punto, ossia dopo aver inserito e salvato il disegno, potremo vedere il risultato nella pagina grafica scegliendo l'opzione 5 dal menu principale ("Vedi grafica disegno").

Dopo l'attivazione della pagina grafica vengono disegnate **dapprima** le linee, **poi** i rettangoli ed **infine** i cerchi. Alla fine, la pressione di un tasto ci riporterà al menu principale.

Per quanto riguarda l'array delle linee (numerate da 1 al puntatore Pl) ognuna di esse viene disegnata con un'istruzione del tipo:

LINE (X1, Y1) - (X2, Y2), 1

...in cui l'ultimo parametro (1) rappresenta il colore.

Nel listato, a causa dei parametri necessari, l'istruzione diventa:

LINE (Li (K, 0), Li (K, 1)) - (Li (K, 2), Li (K, 3)), 1

Stessa cosa per i rettangoli, che sono disegnati sempre tramite l'istruzione Line con opzione "**B**" (se non ricordate come si usano le istruzioni Gw-Basic è il momento di dare un'occhiata al manuale):

LINE (Re (K, 0), Re (K, 1)) - (Re (K, 2), Re (K, 3)), 1, B

L'opzione "B" indica che la figura da disegnare non è una linea ma un rettangolo (**Box**).

Le circonferenze vengono disegnate per ultime, tramite:

CIRCLE (X, Y), R, 1, 0, 6.28, Aspetto

Il numero 1 rappresenta il colore.

Anche qui, a causa dei numerosi parametri, l'istruzione diventa...

CIRCLE (Ce (K, 0), Ce (K, 1)), Ce (K, 2), 1, 0, 6.28, Ce (K, 3)

I parametri "**0**" e "**6.28**" indicano che il cerchio deve essere disegnato interamente, ossia da **0** radianti a **2*pigreco** radianti.

Riducendo opportunamente il valore 6.28 è possibile disegnare archi di cerchio (= suggerimento per eventuale modifica).

Alla fine si aspetta la pressione di un tasto che rimette lo schermo in modo testo.

La routine di disegno non esegue alcun controllo sui parametri inseriti negli array; questi dovranno essere inseriti correttamente dall'utente, pena qualche segnalazione di errore tipo "illegal function call" o simili.

Come correggere

Il nostro capolavoro, però, potrebbe non piacerci, oppure la presenza di qualche errore ci può costringere ad effettuare le dovute correzioni.

Viene quindi in aiuto la provvidenziale opzione "Vedi/Modifica composizione disegno", richiamata da menu principale tramite il tasto 3.

Effettuata la scelta dell'elemento da correggere (tra linee, rettangoli o cerchi), l'array corrispondente all'elemento prescelto viene visualizzato su video in un modo simile a quello dei tabelloni elettronici. Gli elementi del disegno vengono visualizzati **15 per volta**, per problemi di spazio sul video. Sulle ultime due righe compariranno alcune opzioni :

1) Prossimi 15
2) Modifica
3) Cancella
4) Vedi disegno
5) Ritorna al menu di modifica

Premendo 1 potranno essere visualizzati i successivi 15 elementi dell'array.

Con 2 è possibile modificare uno degli elementi inseriti nell'array. MicroCad chiede, in questo caso, il numero dell'elemento da modificare, che ovviamente deve essere compreso tra 1 ed il numero di elementi già inseriti.

Una volta inserito l'indice, il disegno viene nuovamente riproposto in grafica, con l'elemento scelto che **lampeggia** sullo schermo.

Premendo un tasto la visualizzazione grafica termina e MicroCad ripropone gli elementi dell'array chiedendo di inserire i nuovi parametri per l'elemento da modificare.

Se decidiamo di lasciare inalterati i precedenti parametri (perchè, ad esempio, ci siamo sbagliati ad indicare l'elemento) sarà sufficiente inserire la solita sequenza "*,,," e premere Return; altrimenti dovremo inserire i nuovi valori, separati da virgole.

Premendo 3 (Cancella) si potrà cancellare un elemento dell'array. Anche in questo caso verrà chiesto il numero dell'elemento da cancellare, il disegno verrà visualizzato in grafica con l'elemento prescelto che lampeggia.

Terminata la visualizzazione grafica potremo premere "**1**" per cancellare effettivamente l'elemento oppure "**0**" per non cancellarlo e tornare al menu precedente.

Premendo 4 (visualizza disegno) MicroCad visualizzerà il disegno presente in memoria.

Questa opzione è equivalente all'opzione "Vedi grafica disegno" del menu principale.

Premendo 5 torneremo al menu "Vedi/Modifica composizione disegno".

La parte più interessante della modifica e della cancellazione di un elemento del disegno, è quella che consente di farlo lampeggiare. Dapprima viene disegnato l'elemento (con le istruzioni viste prima) con il **colore 1**, poi si aspetta un certo lasso di tempo e quindi si ridisegna lo stesso elemento con il **colore 0** (e quindi lo si cancella); il ciclo ricomincia fino alla pressione di un tasto.

Per il resto, non c'è niente di difficile. L'array contenente i parametri del disegno viene visualizzato tramite semplici Print ed una brevissima routine di formattazione che consente di stampare numeri incolonnati correttamente.

Per modificare un elemento vengono richiesti in Input i numeri componenti l'elemento e quindi sostituiti a quelli vecchi presenti nell'array.

Per cancellarlo, invece, vengono "spostati indietro" di una posizione, nell'array, tutti gli elementi successivi a quello prescelto, mentre il puntatore all'array (Pl, Pr o Pc, a seconda dei casi) viene decrementato di 1.Neanche in questa fase vengono controllati gli inserimenti, ed in particolare le coordinate degli elementi del disegno. Bisogna quindi fare attenzione a ciò che viene inserito.

L'opzione 6 da menu principale (Cancella disegno) permette di cancellare l'intero disegno, oppure tutte le linee, tutti i rettangoli o tutti i cerchi. Il solito menu viene proposto:

0) Tutto
1) Linee
2) Rettangoli
3) Cerchi
4) Ritorna al menu principale

Note per la digitazione del listato:

1) Su Ms-Dos o compatibili:
- Caricare Gw-Basic
- Digitare e salvare il listato
- Farlo partire

Note:

- Nel listato sono utilizzati nomi di variabili lunghi due soli caratteri allo scopo di mantenere la compatibilità con il **Gw-Basic del C/64**; Il "vero" Gw-Basic ne potrebbe accettare molti di più.
- MicroCad è stato sviluppato con **scheda grafica CGA**, e produce esattamente la stessa schermata del C/64. Utilizzando altre schede grafiche (EGA, Hercules, ecc.) i risultati potrebbero cambiare per ciò che riguarda la dimensione dei disegni e l'aspetto delle circonferenze.

2) Su C/64:
- Caricare l'emulatore Gw-Basic
- Digitare e quindi salvare il listato
- Farlo partire

Note:

- L'istruzione Key Off, sul C/64 diventa **Keyoff**
- L'istruzione Width 40, necessaria per portare lo schermo dell'Ms-Dos a 40 colonne, **non** va digitata sul C/64 (pena un Syntax error).
- Le istruzioni del tipo **Print #** e **Input #** del vero Gw-Basic, sul C/64 diventano **Print#** e **Input#** (cancelletto attaccato all'istruzione **Print** o **Input**, senza spazio intermedio); Print **non** può essere abbreviato con il simbolo del punto interrogativo (**?**).
- Per una strana gestione della grafica sul C/64 (dipendente dall'hardware) alcune linee verticali potrebbero non vedersi (specialmente quelle con coordinate orizzontali pari), quindi bisogna regolare le coordinate di conseguenza.
- **Il C/64 è circa 10 volte più lento** (soprattutto in grafica e nei caricamenti da disco) nell'esecuzione del programma rispetto ad un XT compatibile; non aspettatevi miracoli, soprattutto nel caso di disegni complessi.

Il listato potrebbe anche essere digitato prima su C/64 e poi, tramite collegamento via... satellite RS-232, essere trasferito su un Ms-Dos compatibile. Ovviamente ciò vale solo per chi possiede sia le due macchine sia l'interfaccia di comunicazione RS-232 che permette di collegarle (vedi **CCC n. 49** per i dettagli sul collegamento).

Dovremo semplicemente inserire il numero corrispondente all'operazione desiderata e premere return. MicroCad farà il resto.

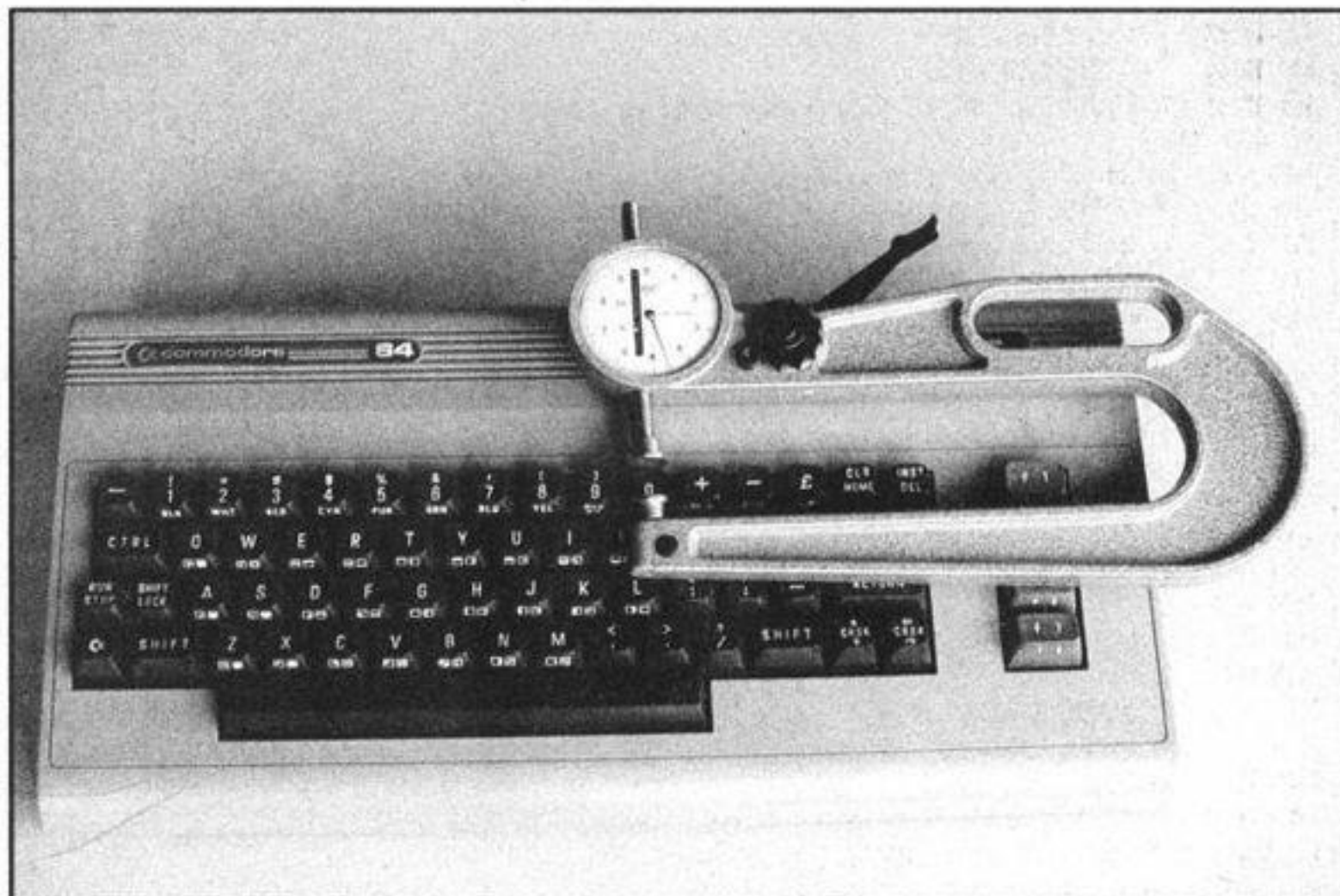
Per cancellare tutto il disegno, MicroCad esegue un'istruzione Run, che quindi cancella tutte le variabili e fa ripartire il programma. Scegliendo le opzioni da 1 a 3, Vengono azzerati solamente i puntatori degli array Pl, Pr o Pc.

Non viene richiesta conferma alla cancellazione: usare con cautela!

Infine, l'opzione "Carica Disegno" da menu principale, permette di caricare da disco un disegno precedentemente salvato con "Salva".

Viene qui proposto lo stesso menu di "Salva Disegno" (vedi sopra) e con la scelta delle opzioni ci si comporta allo stesso modo.

Viene quindi richiesto il nome del disegno da caricare. Sul video, durante il caricamento, verrà visualizzato il tipo di elemento che si sta caricando (Linee, Rettangoli o Cerchi). A fine caricamento, MicroCad tornerà al menu precedente.

Per caricare in memoria un disegno viene utilizzata l'istruzione "parallela" a Print#, cioè Input#. In pratica, vengono aperti i files **"nome.lin"** (linee), **"nome.ret"** (rettangoli) e **"nome.cer"** (cerchi) e, tramite un ciclo While Not Eof... Wend, vengono caricati gli elementi ed inseriti nel corrispondente array, fino alla fine del file. I puntatori degli array vengono quindi aggiornati al corretto numero di elementi caricati.

Come per l'opzione "Salva", anche qui non viene effettuato alcun controllo sul nome del disegno e sulla presenza, o meno, del dischetto nel drive, oppure se il numero di elementi caricati supera 50. Inoltre, se il file prescelto non è presente su disco, MicroCad si "inchioderà" con un messaggio del tipo "File Not Found".

L'ultima opzione del menu principale è la numero 7: Fine MicroCad. Questa fa terminare il programma e tornare in ambiente Gw-Basic.

```
10 REM
11 REM MICRO-CAD
12 REM per Commodore 64+Gwbasic o IBM+Gwbasic
13 REM by Mariani Giancarlo 1990
15 REM
90 KEY OFF: REM ** Sul Commodore64 : KEYOFF **
92 WIDTH 40: REM ** Solo per IBM. Non digitare su C64 !! **
95 REM
100 DIM LI(50, 3): REM Linee (X1,Y1,X2,Y2)
110 DIM RE(50, 3): REM Rettangoli (X1,Y1,X2,Y2)
120 DIM CE(50, 3): REM Cerchi (X,Y,R,Aspect)
125 DIM TE(50, 3): REM Array temporaneo x modifiche,ecc.
130 PL = 0: REM puntatore array linee
140 PR = 0: REM puntatore array rettangoli
150 PC = 0: REM Puntatore array cerchi
160 PT = 0: REM Puntatore array temporaneo
165 T$(1) = " LINEE": T$(2) = " RETTANGOLI": T$(3) = " CERCHI"
170 REM
200 SCREEN 0: CLS
210 PRINT "MICRO-CAD for IBM & C64 GWBASIC"
220 PRINT "by Gianky'90": PRINT
230 PRINT "1 : Carica Disegno"
240 PRINT "2 : Salva Disegno"
250 PRINT "3 : Vedi/modifica composizione disegno"
265 PRINT "4 : Inserimento composizione disegno"
266 PRINT "5 : Vedi grafica disegno"
267 PRINT "6 : Cancellazione disegno"
270 PRINT "7 : Fine CAD": PRINT
280 S$ = "": INPUT "Scegli "; S$
290 IF S$ < "1" OR S$ > "7" THEN GOTO 200
300 IF S$ = "1" THEN GOSUB 3000: GOTO 200
310 IF S$ = "2" THEN GOSUB 2500: GOTO 200
320 IF S$ = "3" THEN GOSUB 1500: GOTO 200
330 IF S$ = "4" THEN GOSUB 1000: GOTO 200
```

```
340 IF S$ = "5" THEN GOSUB 10000: GOSUB 9600: GOTO 200
350 IF S$ = "6" THEN GOTO 2000
360 IF S$ = "7" THEN CLS : PRINT "Fine CAD...": END
500 GOTO 200
1000 REM * Inserimento composizione disegno
1005 CLS : PRINT "INSERIMENTO COMPOSIZIONE DISEGNO"
1010 GOSUB 9500: IF E = 1 THEN GOTO 1000
1020 IF T = 4 THEN RETURN
1030 CLS : PT = 0
1040 ON T GOSUB 9000, 9100, 9200
1050 PRINT PT; T$(T); " PRESENTI": PRINT
1055 PRINT "Per ogni INPUT inserisci : ";
1060 IF T = 1 OR T = 2 THEN PRINT "X1,Y1,X2,Y2"
1070 IF T = 3 THEN PRINT "X,Y,Raggio,Aspetto"
1080 PRINT "(Inserisci '*,,,' per finire)": PRINT
1090 PRINT "Numero "; PT + 1;
1100 INPUT X1$, X2$, X3$, X4$: IF X1$ = "*" THEN GOSUB 9300: GOTO 1000
1105 PT = PT + 1
1110 TE(PT, 0) = VAL(X1$): TE(PT, 1) = VAL(X2$)
1120 TE(PT, 2) = VAL(X3$): TE(PT, 3) = VAL(X4$)
1130 GOTO 1090
1500 REM * Vedi/Modifica composizione disegno
1501 IF PL = 0 AND PC = 0 AND PR = 0 THEN RETURN
505 CLS : PRINT "VEDI/MODIFICA COMPOSIZIONE DISEGNO"
1510 GOSUB 9500: IF E = 1 THEN GOTO 1500
1520 IF T = 4 THEN RETURN
1530 PRINT : PT = 0
1540 ON T GOSUB 9000, 9100, 9200
1550 PRINT PT; T$(T); " PRESENTI": PRINT
1560 FOR X = 1 TO 1000: NEXT X
1565 IF PT = 0 THEN GOTO 1500
1566 I = 1
1567 REM *
1570 IF I > PT THEN I = PT - 15
1575 IF I <= 0 THEN I = 1
1576 IF PT = 0 THEN GOTO 1500
1600 GOSUB 8900
1605 GOSUB 8850
1610 LOCATE 22, 1: PRINT "1 = prossimi 15, 2 = mod., 3 = canc."
1620 LOCATE 23, 1: PRINT "4 = vedi dis., 5 = ritorno ";
1630 S$ = "": INPUT S$: IF S$ < "1" AND S$ > "5" THEN GOTO 1605
1640 IF S$ = "5" THEN GOTO 1500
1650 IF S$ = "1" THEN I = I + 15: GOTO 1570
1655 IF S$ = "4" THEN GOSUB 10000: GOSUB 9600: GOTO 1570
1660 GOSUB 8850
1680 LOCATE 22, 1: N$ = "": INPUT "Numero (0=ritorno) "; N$
1685 N = VAL(N$): IF N = 0 THEN GOTO 1570
1690 IF N < 1 OR N > PT THEN BEEP: GOTO 1660
1700 GOSUB 35000: REM lampeggio oggetto prescelto
1710 IF S$ = "2" THEN GOTO 1850
1720 REM * cancella oggetto
1730 GOSUB 8900: GOSUB 8850
1735 LOCATE 22, 1: PRINT "1 = cancella, 0 = non cancella";
1740 S$ = "": INPUT S$: IF S$ <> "1" THEN GOTO 1570
1750 IF N >= PT THEN PT = PT - 1: GOSUB 9300: GOTO 1570
1760 FOR X = N + 1 TO PT
1770   TE(X - 1, 0) = TE(X, 0): TE(X - 1, 1) = TE(X, 1)
1780   TE(X - 1, 2) = TE(X, 2): TE(X - 1, 3) = TE(X, 3)
1790 NEXT X
1800 PT = PT - 1: GOSUB 9300: GOTO 1570
1850 REM * modifica oggetto
1860 GOSUB 8900: GOSUB 8850
1865 LOCATE 22, 1
1870 IF T = 1 OR T = 2 THEN PRINT "X1,Y1,X2,Y2 ";
1880 IF T = 3 THEN PRINT "X,Y,Raggio,Asp. ";
1885 PRINT "n."; N; "('*' = fine)"
1890 LOCATE 23, 1: INPUT X1$, X2$, X3$, X4$
1895 IF X1$ = "*" THEN GOTO 1570
1900 TE(N, 0) = VAL(X1$): TE(N, 1) = VAL(X2$)
1910 TE(N, 2) = VAL(X3$): TE(N, 3) = VAL(X4$)
1920 GOSUB 9300: GOTO 1570
2000 REM * Cancellazione disegno
2010 IF PR = 0 AND PL = 0 AND PC = 0 THEN GOTO 200
2015 CLS : PRINT "CANCELLAZIONE DISEGNO"
2020 GOSUB 9400: IF E = 1 THEN GOTO 2000
2030 IF T = 4 THEN GOTO 200
2040 IF T = 0 THEN RUN
2050 IF T = 1 THEN PL = 0
2060 IF T = 2 THEN PR = 0
2070 IF T = 3 THEN PC = 0
2080 GOTO 200
2500 REM * salvataggio disegno
2501 REM
2502 REM *  NB per il Commodore 64 :  *
2503 REM * "PRINT #" diventa "PRINT#" *
2504 REM * "INPUT #" diventa "INPUT#" *
2505 REM
2510 IF PR = 0 AND PL = 0 AND PC = 0 THEN RETURN
2520 CLS : PRINT "SALVATAGGIO DISEGNO"
2530 GOSUB 9400: IF E = 1 THEN GOTO 2500
2540 IF T = 4 THEN RETURN
2550 PRINT : N$ = "":
2552 INPUT "Nome disegno "; N$: IF N$ = "" THEN GOTO 2500
2555 IF LEN(N$) > 8 THEN N$ = LEFT$(N$, 8)
2560 PRINT : PRINT "Salvataggio in corso..."
2570 IF (T <> 0 AND T <> 1) OR (PL = 0) THEN GOTO 2640
2575 REM * salvataggio linee
2580 OPEN N$ + ".lin" FOR OUTPUT AS #1
2585 PRINT #1, "LINEE DEL DISEGNO: "; N$
2590 FOR X = 1 TO PL
2600   PRINT #1,, STR$(LI(X, 0)); ","; STR$(LI(X, 1)); ",";
2610   PRINT #1, STR$(LI(X, 2)); ","; STR$(LI(X, 3))
2620 NEXT X
2630 CLOSE #1
2640 IF (T <> 0 AND T <> 2) OR (PR = 0) THEN GOTO 2710
2645 REM * salvataggio rettangoli
2650 OPEN N$ + ".ret" FOR OUTPUT AS #1
2655 PRINT #1, "RETTANGOLI DEL DISEGNO: "; N$
2660 FOR X = 1 TO PR
2670   PRINT #1, STR$(RE(X, 0)); ","; STR$(RE(X, 1)); ",";
2680   PRINT #1, STR$(RE(X, 2)); ","; STR$(RE(X, 3))
2690 NEXT X
2700 CLOSE #1
2710 IF (T <> 0 AND T <> 3) OR (PC = 0) THEN GOTO 2500
2720 REM * salvataggio cerchi
2730 OPEN N$ + ".cer" FOR OUTPUT AS #1
2735 PRINT #1, "CERCHI DEL DISEGNO: "; N$
2740 FOR X = 1 TO PC
2750   PRINT #1, STR$(CE(X, 0)); ","; STR$(CE(X, 1)); ",";
2760   PRINT #1, STR$(CE(X, 2)); ","; STR$(CE(X, 3))
2770 NEXT X
2780 CLOSE #1
2790 GOTO 2500
3000 REM * Caricamento disegno
3010 CLS : PRINT "CARICAMENTO DISEGNO"
3020 GOSUB 9400: IF E = 1 THEN GOTO 3000
3030 IF T = 4 THEN RETURN
3040 PRINT : N$ = "": INPUT "Nome disegno "; N$: IF N$ = "" THEN GOTO 3000
3050 PRINT : PRINT "Caricamento in corso..."
3060 IF (T <> 0 AND T <> 1) THEN GOTO 3140
3070 REM * Caricamento linee
3080 PL = 0: OPEN N$ + ".lin" FOR INPUT AS #1
3085 INPUT #1, A$: PRINT A$
3090 WHILE NOT EOF(1)
3100   PL = PL + 1: INPUT #1, X1$, X2$, X3$, X4$
3105   LI(PL, 0) = VAL(X1$): LI(PL, 1) = VAL(X2$)
3110   LI(PL, 2) = VAL(X3$): LI(PL, 3) = VAL(X4$)
3120 : WEND
3130 CLOSE #1
3140 IF (T <> 0 AND T <> 2) THEN GOTO 3210
3145 REM * Caricamento rettangoli
3150 PR = 0: OPEN N$ + ".ret" FOR INPUT AS #1
```

```
3155 INPUT #1, A$: PRINT A$
3160 WHILE NOT EOF(1)
3170   PR = PR + 1: INPUT #1, X1$, X2$, X3$, X4$
3175   RE(PR, 0) = VAL(X1$): RE(PR, 1) = VAL(X2$)
3180   RE(PR, 2) = VAL(X3$): RE(PR, 3) = VAL(X4$)
3190 : WEND
3200 CLOSE #1
3210 IF (T <> 0 AND T <> 3) THEN GOTO 3000
3220 REM * Caricamento cerchi
3230 PC = 0: OPEN N$ + ".cer" FOR INPUT AS #1
3235 INPUT #1, A$: PRINT A$
3240 WHILE NOT EOF(1)
3250   PC = PC + 1: INPUT #1, X1$, X2$, X3$, X4$
3255   CE(PC, 0) = VAL(X1$): CE(PC, 1) = VAL(X2$)
3260   CE(PC, 2) = VAL(X3$): CE(PC, 3) = VAL(X4$)
3270 : WEND
3280 CLOSE #1
3290 GOTO 3000
8850 REM * cancella le righe 22 e 23 del video
8860 LOCATE 22, 1: PRINT SPACE$(39);
8870 LOCATE 23, 1: PRINT SPACE$(39);
8880 RETURN
8900 REM * Visualizza l'array temporaneo da I a I+15
8901 CLS : IF T = 1 OR T = 2 THEN PRINT "              X1    Y1    X2    Y2"
8902 IF T = 3 THEN PRINT "              X     Y     R     Asp"
8904 PRINT : R = I
8905 IF R > PT THEN RETURN
8910 PRINT "Numero "; RIGHT$("   " + STR$(R), 3); ":"; : GOSUB 8950
8930 R = R + 1: IF R < I + 15 THEN GOTO 8905
8940 RETURN
8950 REM * Visualizza una riga di array temporaneo. Indice = r
8955 FOR K = 0 TO 3
8956   X = TE(R, K)
8970   X$ = RIGHT$("     " + STR$(X), 5)
8975   PRINT X$; " ";
8976 NEXT K: PRINT
8980 RETURN
9000 REM * Travasa l'array Linee nell'array Temporaneo
9005 PT = PL
9010 IF PL = 0 THEN RETURN
9020 FOR K = 1 TO PL
9030   TE(K, 0) = LI(K, 0): TE(K, 1) = LI(K, 1)
9035   TE(K, 2) = LI(K, 2): TE(K, 3) = LI(K, 3)
9040 NEXT K
9060 RETURN
9100 REM * Travasa l'array Rettangoli nell'array Temporaneo
9105 PT = PR
9110 IF PR = 0 THEN RETURN
9120 FOR K = 1 TO PR
9130   TE(K, 0) = RE(K, 0): TE(K, 1) = RE(K, 1)
9135   TE(K, 2) = RE(K, 2): TE(K, 3) = RE(K, 3)
9140 NEXT K
9160 RETURN
9200 REM * Travasa l'array Cerchi nell'array Temporaneo
9205 PT = PC
9210 IF PC = 0 THEN RETURN
9220 FOR K - 1 TO PC
9230   TE(K, 0) = CE(K, 0): TE(K, 1) = CE(K, 1)
9235   TE(K, 2) = CE(K, 2): TE(K, 3) = CE(K, 3)
9240 NEXT K
9260 RETURN
9300 REM * Travasa l'array Temporaneo nell'array Linee, cerchi o rettangoli
9302 IF T = 1 THEN PL = PT
9304 IF T = 2 THEN PR = PT
9306 IF T = 3 THEN PC = PT
9310 IF PT = 0 THEN RETURN
9320 FOR K = 1 TO PT
9330   IF T = 1 THEN LI(K, 0) = TE(K, 0): LI(K, 1) = TE(K, 1)
9331   IF T = 1 THEN LI(K, 2) = TE(K, 2): LI(K, 3) = TE(K, 3)
9332   IF T = 2 THEN RE(K, 0) = TE(K, 0): RE(K, 1) = TE(K, 1)
9333   IF T = 2 THEN RE(K, 2) = TE(K, 2): RE(K, 3) = TE(K, 3)
9334   IF T = 3 THEN CE(K, 0) = TE(K, 0): CE(K, 1) = TE(K, 1)
9335   IF T = 3 THEN CE(K, 2) = TE(K, 2): CE(K, 3) = TE(K, 3)
9340 NEXT K
9360 RETURN
9400 REM * Scelta tra tutti,linee,cerchi o rettangoli
9410 DN$ = "0"
9420 PRINT : PRINT "0 : Tutto"
9430 GOTO 9502
9500 REM * Scelta tra Linee,Cerchi o rettangoli
9501 DN$ = "1": PRINT
9502 E = 0
9510 PRINT "1 : Linee"
9520 PRINT "2 : Rettangoli"
9530 PRINT "3 : Cerchi"
9535 PRINT "4 : Ritorno al menu principale": PRINT
9540 S$ = "": INPUT "Scegli "; S$: IF S$ < DN$ OR S$ > "4" THEN E = 1: RETURN
9550 T = VAL(S$)
9560 PRINT
9570 RETURN
9600 REM ** Aspetta la pressione di un tasto e mette in testo
9610 A$ = INKEY$: IF A$ = "" THEN GOTO 9610
9620 A$ = INKEY$: SCREEN 0
9630 RETURN
10000 REM * Disegna *
10010 REM * INPUT: Arrays Li(),Re(),Ce() e puntatori Pl,Pr,Pc *
10020 SCREEN 1: CLS
10025 IF PL = 0 THEN GOTO 10070
10027 REM * Disegna linee
10030 FOR K = 1 TO PL
10040   C = 1: GOSUB 30000
10060 NEXT K
10070 IF PR = 0 THEN GOTO 10130
10080 REM * Disegna rettangoli
10090 FOR K = 1 TO PR
10100   C = 1: GOSUB 31000
10110 NEXT K
10120 IF PC = 0 THEN GOTO 11000
10125 REM * Disegna cerchi
10130 FOR K = 1 TO PC
10140   C = 1: GOSUB 32000
10150 NEXT K
11000 RETURN
30000 REM ** Disegna o cancella una linea
30005 REM ** K=indice array Li(), C = colore
30010 X1 = LI(K, 0): Y1 = LI(K, 1): X2 = LI(K, 2): Y2 = LI(K, 3)
30020 LINE (X1, Y1)-(X2, Y2), C
30030 RETURN
31000 REM ** Disegna o cancella un rettangolo
31005 REM ** K=indice array Re(), C = colore
31010 X1 = RE(K, 0): Y1 = RE(K, 1): X2 = RE(K, 2): Y2 = RE(K, 3)
31020 LINE (X1, Y1)-(X2, Y2), C, B
31030 RETURN
32000 REM ** Disegna o cancella un cerchio
32005 REM ** K=indice array Re(), C = colore
32010 X = CE(K, 0): Y = CE(K, 1): RA = CE(K, 2): AP = CE(K, 3)
32020 CIRCLE (X, Y), RA, C, 0, 6.28, AP
32030 RETURN
35000 REM * fa lampeggiare l'elemento che si vuole
35010 REM * modificare o cancellare
35015 REM * t = 1: linee 2:rettangoli 3:cerchi
35016 REM * k = elemento da far lampeggiare
35020 GOSUB 10000: K = N
35030 C = 0
35040 IF T = 1 THEN GOSUB 30000
35050 IF T = 2 THEN GOSUB 31000
35060 IF T = 3 THEN GOSUB 32000
35070 FOR U = 1 TO 500: NEXT U
35080 A$ = INKEY$
35090 IF A$ = "" THEN C = 1 - C: GOTO 35040
35100 SCREEN 0: A$ = INKEY$
35110 RETURN
```

SPECIALE

Corso completo di linguaggio "C" per C64, Amiga e PC

Questo articolo è il primo di una lunga serie destinata a tutti coloro che hanno deciso di passare al linguaggio più diffuso e più portabile tra quelli presenti nel mondo dell'informatica. Particolarmente adatto ai possessori di Personal Computer Ibm compatibili ma grazie allo standard raggiunto per il C, indicato anche per Amiga e C64, il C a tutt'oggi rappresenta il linguaggio del futuro

L'autore di questa serie è Giancarlo Mariani, già noto ai lettori di Commodore Computer Club per la realizzazione di progetti altamente impegnativi quali l'emulatore Ms-Dos e Gw-Basic, ampiamente utilizzato sulle pagine di questa rivista, e l'emulatore Turbo Pascal, entrambi per C64

Il programma proposto esegue le classiche quattro operazioni su due numeri, richiesti da tastiera, con una piccola aggiunta, ossia un controllo di divisione per zero, che, per esperienza tratta da numerosi anni di studio, sappiamo essere impossibile.

❑❒

Esame del listato

Esaminiamo ora passo-passo il programma scritto in C, che chiameremo **Operaz.C**, confrontandolo con il corrispondente listato **Gw-Basic** (che gira anche sul **C/64** dotato del nostro emulatore!) chiamato **Operaz.Bas**.

Il lettore, pertanto, non deve fare altro che leggere il testo che segue, confrontandolo con i listati che compaiono in queste pagine.

/*

La prima riga del listato in **C** contiene un carattere **slash** (barra) seguito da un asterisco (/*).

I due segni marcano l'apertura di un commento; ciò significa che, da quel punto in poi, il testo che eventualmente compare è considerato ininfluente ai fini dell'elaborazione vera e propria, anche se si scrive più di una riga (nel listato sono presenti quattro righe di commento).

La chiusura del commento viene effettuata con gli stessi caratteri dell'apertura, pur se posti al contrario: prima l'asterisco e poi lo slash (*/). Il corrispondente in Basic, come sappiamo, si ottiene con l'istruzione **Rem**, limitata ad una sola per ogni riga per ciascun commento.

void main()

In C il programma principale viene considerato come una qualsiasi subroutine, e pertanto bisogna comunicare al linguaggio il punto esatto in cui inizia il main (=principale) con l'istruzione, appunto, **main**. Le due parentesi tonde poste dopo l'istruzione indicano che, nel nostro caso specifico, main non ha argomenti; ricordiamo, infatti, che il C permette di "passare" argomenti ad un programma richiamato da Ms-Dos; tale concetto sarà più chiaramente sviluppato sui prossimi numeri della nostra pubblicazione.

L'istruzione void, che troviamo prima di main, indica che il programma, o meglio la subroutine di nome main, non ha alcun valore "di ritorno". Il parametro **void** serve, in C, per differenziare le subroutine dalle funzioni, che, come vedremo in un prossimo futuro, presentano, al contrario, valori di ritorno. La parentesi graffa aperta ({), presente sul rigo successivo a quello che contiene l'istruzione main, "apre" effettivamente il programma principale, e potrebbe essere l'equivalente di un **Begin** in Pascal. In Basic, come è noto, il main program non ha bisogno di essere dichiarato.

Le variabili

Subito dopo la dichiarazione del main, vi sono le dichiarazioni delle variabili, che quindi restano **interne** al main e visibili solo da esso e non da altre eventuali subroutines o funzioni.

In C, a differenza del Basic, è obbligatorio dichiarare tutte le variabili usate nel programma, pena la

segnalazione di errore da parte del compilatore. Nel programma di queste pagine sono dichiarate **V1** e **V2** come variabili floating point (in virgola mobile), **Scelta** come variabile intera e **Ch** come variabile stringa di un solo carattere. Tale pignoleria contrasta con l'assoluta libertà offerta dal Basic, che non obbliga a dichiarare le variabili.

L'interprete Basic, infatti, considera tutte le variabili come floating point, tranne quelle che finiscono con il punto esclamativo (**!**), che sono considerate intere, e quelle che finiscono con il segno del dollaro ($), le famose stringhe.

Do

Questa istruzione, e la consecutiva parentesi graffa aperta, segna l'inizio di un ciclo che sarà chiuso, come vedremo in seguito, dall'istruzione while.

clrscr

Semplicemente cancella lo schermo. Le due parentesi tonde indicano che l'istruzione non ha parametri, ma sono obbligatorie, altrimenti C non esegue l'istruzione. In Basic, l'equivalente di clrscr è **CLS**.

Printf

Lo scopo delle istruzioni printf poste subito dopo clrscr è facilmente intuibile. E' l'equivalente del **Print** del Basic. In questo caso printf stampa il testo contenuto tra le successive coppie di virgolette. Gli ultimi due caratteri (**\n**) impongono al compilatore di andare a capo dopo aver stampato il messaggio. Il carattere slash (\) posto in una qualsiasi posizione di una stringa, informa C che il carattere posto subito dopo deve esser considerato come carattere di controllo e non come "testo" da stampare.

Nel nostro caso, **n** corrisponde al carattere **Ascii 13** (i tasti **Ctrl** ed **n** provocano, infatti, un ritorno carrello), e quindi la sequenza **\n** inserisce un **Cr** nella stringa.

Il Basic, come sappiamo, va a capo automaticamente dopo ogni istruzione Print, a meno che non si metta il punto e virgola subito dopo.

Scanf

E' l'istruzione equivalente all'**Input** del Basic, ma il suo uso è totalmente diverso. Ogni volta che si stampa o si legge una variabile (tramite **printf** o **scanf**), il compilatore C ha bisogno di essere informato del tipo della variabile in esame. Il compito viene effettuato tramite caratteri di controllo posti nelle istruzioni di lettura o di stampa; nel nostro caso, **%f** e **%f**. Il segno di percentuale indica che il carattere posto subito dopo è un carattere di controllo di formato, che, nel caso specifico (**f**) è del tipo floating point.

La stringa di formato **%f, %f** significa letteralmente: leggi **due** valori del tipo floating point, separati da una virgola.

Altri caratteri di controllo di formato sono **d** (intero), **c** (carattere), **s** (stringa).

A questo punto, ovviamente, dobbiamo dire al compilatore a quali variabili assegnare i valori numerici che introdurremo da tastiera durante l'elaborazione del programma. Le variabili **&V1** e **&V2** completano, quindi, l'istruzione scanf e **devono** essere dello stesso tipo dichiarato nella stringa di controllo (in questo caso floating point), altrimenti C potrebbe fornire risultati imprevedibili.

Rimane da spiegare la presenza del carattere et (**&**) posto davanti ai nomi delle variabili.

Il C, quando occorre assegnare un valore ad una qualsiasi variabile, ha bisogno, come "riferimento", **non** del nome della variabile ma del suo **indirizzo** in memoria. In pratica, con l'espressione &V1, &V2 vengono comunicati, all'istruzione scanf, gli indirizzi delle variabili V1 e V2, in modo che C sappia dove trascrivere i valori introdotti. In pratica l'operatore & restituisce l'indirizzo della variabile stessa.

In Basic, come sappiamo, la gestione delle variabili è totalmente automatizzata, nel senso che un banale Input V1, V2 risolve rapidamente il problema.

Gotoxy

E' l'equivalente di **Locate** del Basic; sposta il cursore alle coordinate indicate.

Le successive cinque istruzioni **printf** preparano la maschera del menu. La successiva istruzione **do** (seguita da una parentesi graffa aperta come quella prima vista) segna l'apertura di un nuovo ciclo, che quindi resta interno al primo. La successiva **scanf** legge un numero intero (notare la stringa di formato **%d**) e lo assegna alla variabile **Scelta** (parametro **&Scelta**).

If

Subito dopo è presente un'istruzione **if**, che permette di stampare un messaggio di errore in caso di scelta errata.

Le differenze, rispetto alla equivalente IF del Basic, sono le seguenti:

- Le condizioni vanno obbligatoriamente poste tra parentesi.
- Non bisogna inserire **Then** (come nel Basic) dopo le condizioni di if.
- Gli operatori **And** e **Or**, noti nel Basic, sono sostituiti, in C, rispettivamente da **&&** e ||; questi ultimi due segni corrispondono al carattere Ascii **124** e, se sul vostro computer Ms-Dos non è presente il tasto specifico, si possono visualizzare, appunto, tenendo premuto il tasto Alt mentre si pigiano in successione i tasti **1**, **2** e **4**. Rilasciando il tasto Alt comparirà il segno |.

L'ultimo ciclo **do** viene chiuso tramite **while**, che impone eventualmente la ripetizione del ciclo fino a che **Scelta** resta minore di 1 oppure maggiore di 4 (Scelta errata).

In Basic, come si può notare dai corrispondenti listati, l'elaborazione viene eseguita da un ciclo **While... Wend**.

Perchè un compilatore

Tutti i computer attualmente in circolazione, dalla più piccola calcolatrice tascabile al più complesso maxicomputer, sono in grado di comprendere, direttamente, solo il linguaggio del loro processore, denominato **linguaggio macchina** (l.m.). Ciò significa, (come spiegato in quasi tutti i numeri di C.C.C.) che ogni processore contiene, al suo interno, una specie di interprete, di tipo **hardware**, che è in grado di capire una limitata serie di istruzioni, appunto il linguaggio macchina.

Questo linguaggio, essendo compreso direttamente dal processore, risulta essere velocissimo.

Lo svantaggio del l.m. è quello di essere poco chiaro e leggibile, e di dover convivere con parecchi numeri binari, esadecimali, locazioni di memoria, ecc.

Per questa ragione il l.m. è poco digeribile, soprattutto dai principianti.

Per ovviare al problema sono stati sviluppati numerosi linguaggi definiti ad **alto livello**, che consentono, con poche e chiare istruzioni, di fare ciò che, in l.m, richiederebbe decine di righe di programmazione.

Ovviamente questi linguaggi non sono direttamente comprensibili dal microprocessore, ma occorre uno "strumento" che altro non è se non un programma traduttore (compilatore od interprete) in grado di tradurre le istruzioni, impartite ad alto livello, in istruzioni l.m. (**basso livello**).

La traduzione richiede, però, tempo aggiuntivo di elaborazione, per cui i linguaggi ad alto livello sono generalmente più lenti del l.m. "puro".

Recentemente un linguaggio ad alto livello ha avuto una larghissima diffusione grazie all'implementazione su piccoli computer quali **Amiga**, **Ms-Dos** e, sia pure in versione sensibilmente ridotta ed inefficace, sul **Commodore 64**.

Il linguaggio in questione è l'ormai famoso **C**.

C è un linguaggio compilato, ossia prima bisogna scrivere il programma "sorgente" con un qualsiasi editor di testi, e poi compilarlo prima di farlo girare.

Il linguaggio unisce la semplicità della programmazione ad alto livello alla velocità di esecuzione, che si avvicina sensibilmente a quella tipica del l.m, ed in ogni caso è di molte volte superiore a quella di un corrispondente programma scritto in altri linguaggi interpreti, tra cui, ad esempio, il **Basic**. Spiegare le caratteristiche di un particolare linguaggio ad una persona che non lo conosce assolutamente non è cosa facile, per cui abbiamo preso la decisione di presentare un semplicissimo esempio di programma in C, confrontato con un medesimo programma scritto in Basic (che si suppone conosciuto da tutti...), in modo da iniziare a comprendere le differenze sostanziali tra i due linguaggi e muovere i primi passi con il C.

Le successive istruzioni **if** determinano il tipo di operazione da compiere.

Per queste ultime c'è da tener conto di alcune considerazioni:

- Il doppio segno di uguale (==) posto in ogni istruzione if non è un errore di stampa, ma un'altra caratteristica di C. Il C, come il Pascal ma a differenza del Basic, differenzia un **assegnamento** da un **controllo**.

Ad esempio, in Basic per assegnare alla variabile **A** il valore **0** oppure per controllare che a sia uguale a 1, dovremmo scrivere, rispettivamente...

a = 0

...e...

IF a = 1

In C i due operatori vengono differenziati tra di loro, per cui la prima operazione viene indicata, come in Basic, con **a = 0**, mentre la seconda diventa **if (a==1)**. Se, per errore, in C scriviamo un'istruzione del tipo if (a=1), il compilatore **non** segnala errore, ma i risultati sono catastrofici per il nostro programma. C, infatti, considera l'istruzione come un assegnamento; ne consegue che dapprima assegna alla variabile **a** il valore **1**, e poi controlla che **a** sia diversa da **0**.

Questo è uno dei più frequenti (ed apparentemente inspiegabili) errori nei programmi C scritti dai principianti (e non...).

Le quattro istruzioni **printf**, poste dopo le altrettante **if**, eseguono e stampano il risultato delle operazioni.

Anche printf, come visto prima per la scanf, quando deve stampare variabili ha bisogno del formato. In questo caso, dato che le variabili sono tutte floating point, il formato sarà sempre **%f**.

Printf, comunque, a differenza di scanf, non richiede l'indirizzo della variabile, ma il valore della stessa, per cui scriveremo il nome della variabile senza il segno **&** posto davanti.

L'operatore punto esclamativo + uguale (**!=**) posto nella penultima istruzione if è l'equivalente del "diverso da" (<>) del Basic. In pratica...

if (V2 != 0)

... significa: se V2 è diverso da 0.

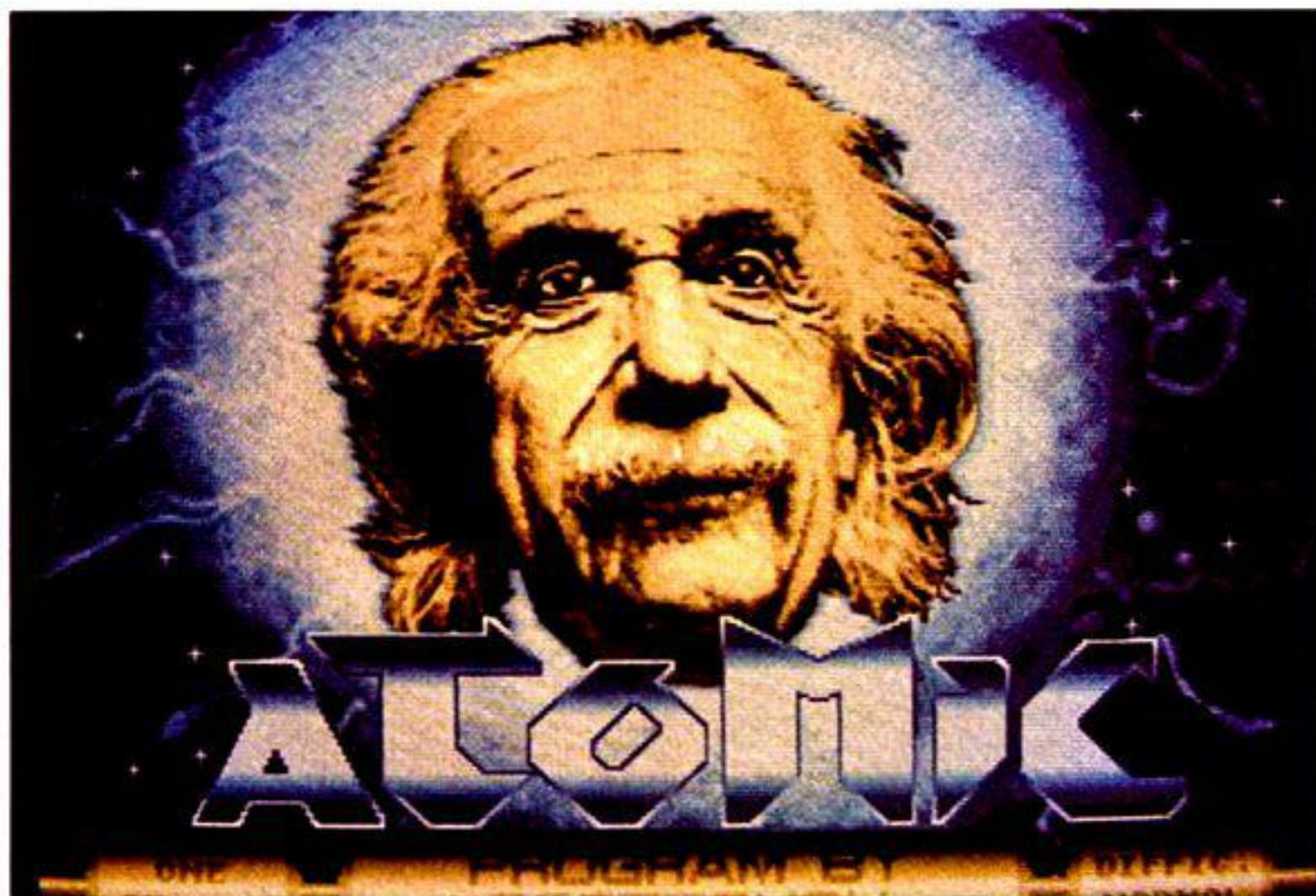
In Basic le istruzioni IF per svolgere i compiti appena esaminati, sono abbastanza simili alle corri-

Commodore
COMPUTER
CLUB
La rivista degli utenti di sistemi Commodore

recensioni

ATOMIX

E' addirittura possibile imparare la chimica con questo gioco che richiede abilità

Computer: Amiga inespanso
Tipo: Arcade Puzzle
Gestione: Joystick
Softhouse: Thalion

La **Thalion** è una giovanissima softhouse tedesca e questo è il suo primo prodotto per Amiga. Un puzzle game originale, vagamente simile a **Pipemania**.

Il gioco

Il gioco consiste in **30** livelli; in ciascuno di essi bisogna spostare adeguatamente degli atomi per formare la molecola di una sostanza chimica.
La prima è la semplicissima acqua, costituita da due atomi di idrogeno ai lati di un atomo di ossigeno.
Per spostare gli atomi si usa il joystick, che li spinge ortogonalmente sinchè non raggiungono una parete od un altro atomo.
Bisogna quindi stare attenti a non bloccarli ed a trovare la giusta sequenza per collegarli tra loro nel minore tempo possibile.
Minore è la quantità di tempo impiegato per formare la molecola, maggiore è il punteggio.
Ovviamente è previsto un tempo limite e, fallito il completamento di un livello, si deve ricominicare daccapo ogni volta. Procedendo, le molecole diventano sempre più complesse ed il campo di gioco più tortuoso. In effetti, si tratta di un programma che potrebbe insegnare qualche rudimento di chimica dei composti!

La tecnica

La grafica è bidimensionale, molto colorata ma senza alcun preziosismo particolare. Lo scorrimento delle molecole è velocissimo, ma l'interfaccia con l'utente banale. Si sarebbe potuto vivacizzare con poco sforzo il campo di gioco, anche perchè la musichetta di sottofondo non è nulla di particolarmente eccitante.

Il voto

Idea simpatica e, in parte, originale. La programmazione risulta scarna, ma la sufficienza è abbondantemente meritata. 6+.

CASTLE MASTER

Un ottimo gioco, che simula in modo preciso il movimento del giocatore

Computer: Amiga inespanso
Tipo: Adventure animata
Gestione: Mouse, Tastiera
Softhouse: Domark

Desiderate fronteggiare un drago di trenta metri od aggirarvi per il tenebroso castello di un mago malvagio? Questa è l'adventure, della serie "grafica quasi arcade", che fa per voi!

Il gioco

Il primo gioco di questo genere fu **Driller** della *Freescape*, nel 1988: una avventura in piena regola, con stanze da esplorare, puzzle da risolvere, oggetti da collezionare e utilizzare correttamente eccetera.Però aveva grafica "solida" che scorreva in tempo reale simulando il movimento del giocatore. Ora, come allora, lo schermo riproduce esattamente, con ovvi limiti, ciò che vede il giocatore mentre cammina, corre, gira la testa, compresi gli altri personaggi in movimento.

Il castello, chiamato **Castle Eternity**, racchiude un prigioniero da liberare e comprende quattro torri, ciascuna con tre livelli. Il primo compito è comunque quello di sopravvivere, ma non è facile: non siamo immortali e dobbiamo mangiare (trovando il cibo) e riposare al sicuro.

Vi sono ovviamente numerosissimi puzzle: oggetti strani, problemi da risolvere, personaggi da "manipolare" adeguatamente affinchè ci aiutino invece che trasformarci in ranocchi, eccetera.

La tecnica

La tecnica di rappresentazione Freescape è stata affinata notevolmente ed i movimenti, in due velocità (camminata e corsa) sono più fluidi che mai, molto docili al mouse.

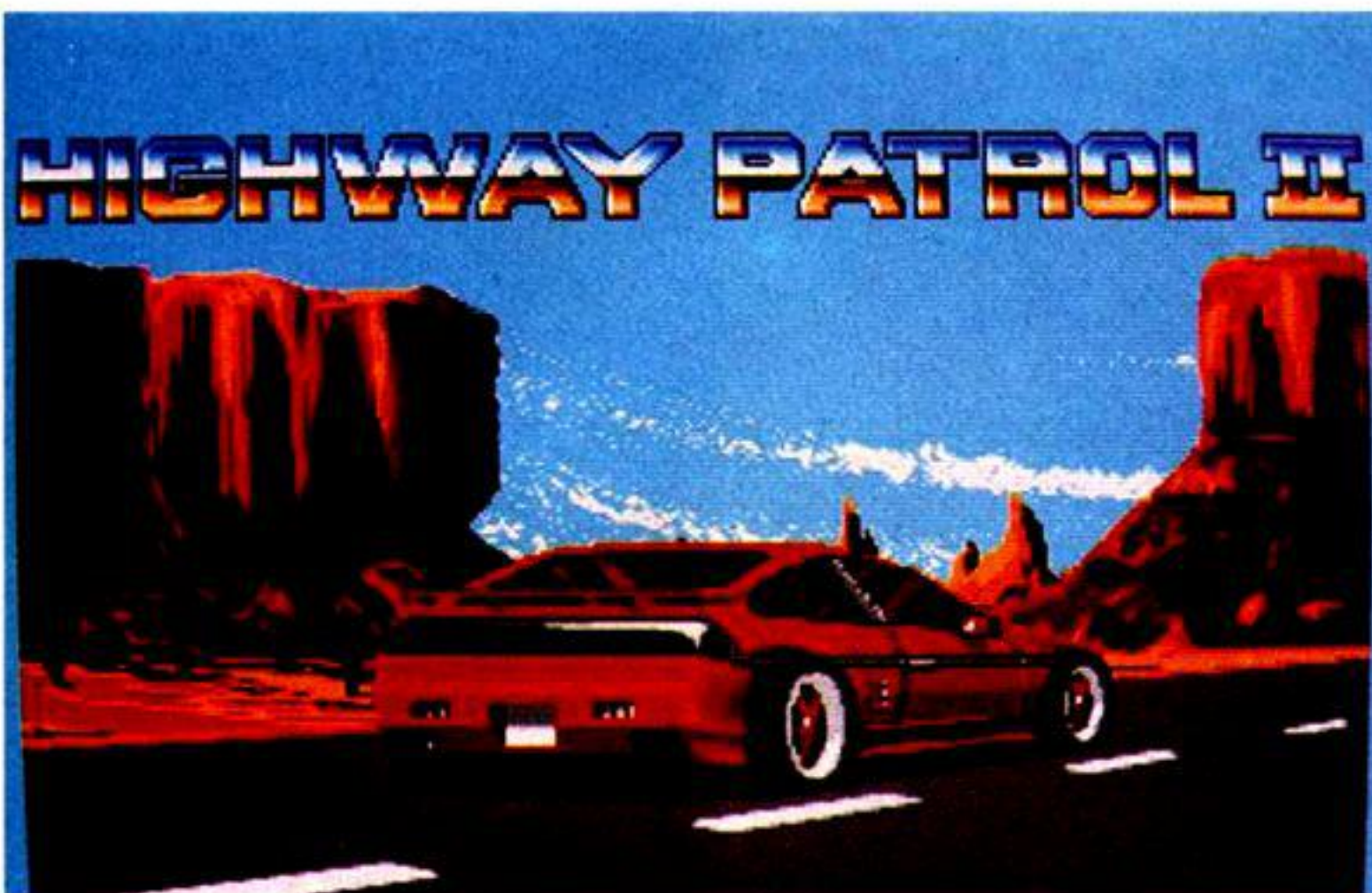
Per ravvivare le scene, i colori predominanti nella grafica di sfondo sono il grigio ed il blu, con un effetto meno spettacolare di quanto si vedeva in Eclipse. Gli effetti sonori sono stati invece piuttosto trascurati.

Il voto

Moltissimi incentivi, ottima grafica ed interazione. Consigliato. 9-.

HIGHWAY PATROL II

Siete un poliziotto terribile: sparate contro tutti gli automobilisti indisciplinati!

Computer: Amiga inespanso
Tipo: Simulatore auto
Gestione: Joystick
Softhouse: Infogrames

Non c'entra nulla con l'originale **Highway Patrol** della Vortex, ma il gioco risulta, comunque, decente.

Il gioco

Si ri-vestono gli scomodi panni di un poliziotto che svolge il servizio di pattugliamento stradale. Stando a bordo della propria macchina, bisogna sparare democraticamente, servendosi di un collimatore elettronico posto sullo schermo, a tutte le macchine che infrangono i limiti di velocità (e noi ci lamentiamo dell'Autovelox...). Inizialmente si sceglie, da un archivio, l'immagine del criminale "wanted", il cui valore è, ovviamente, proporzionale alla sua abilità di guida.

Per trovare poi i nemici dell'ordine pubblico, si deve consultare l'approssimativo radar posto a bordo (ricorda quello di **Defender**) che, con rozzi ed agitati puntini, indica i nostri bersagli. Correre ad alta velocità, tenendo d'occhio il radar, non è affatto semplice.

Oltretutto si dovrebbe trovare il tempo di consultare le coordinate, in alto sullo schermo, perchè paragonando le nostre e quelle dell'avversario si può calcolare distanza e strada da seguire...

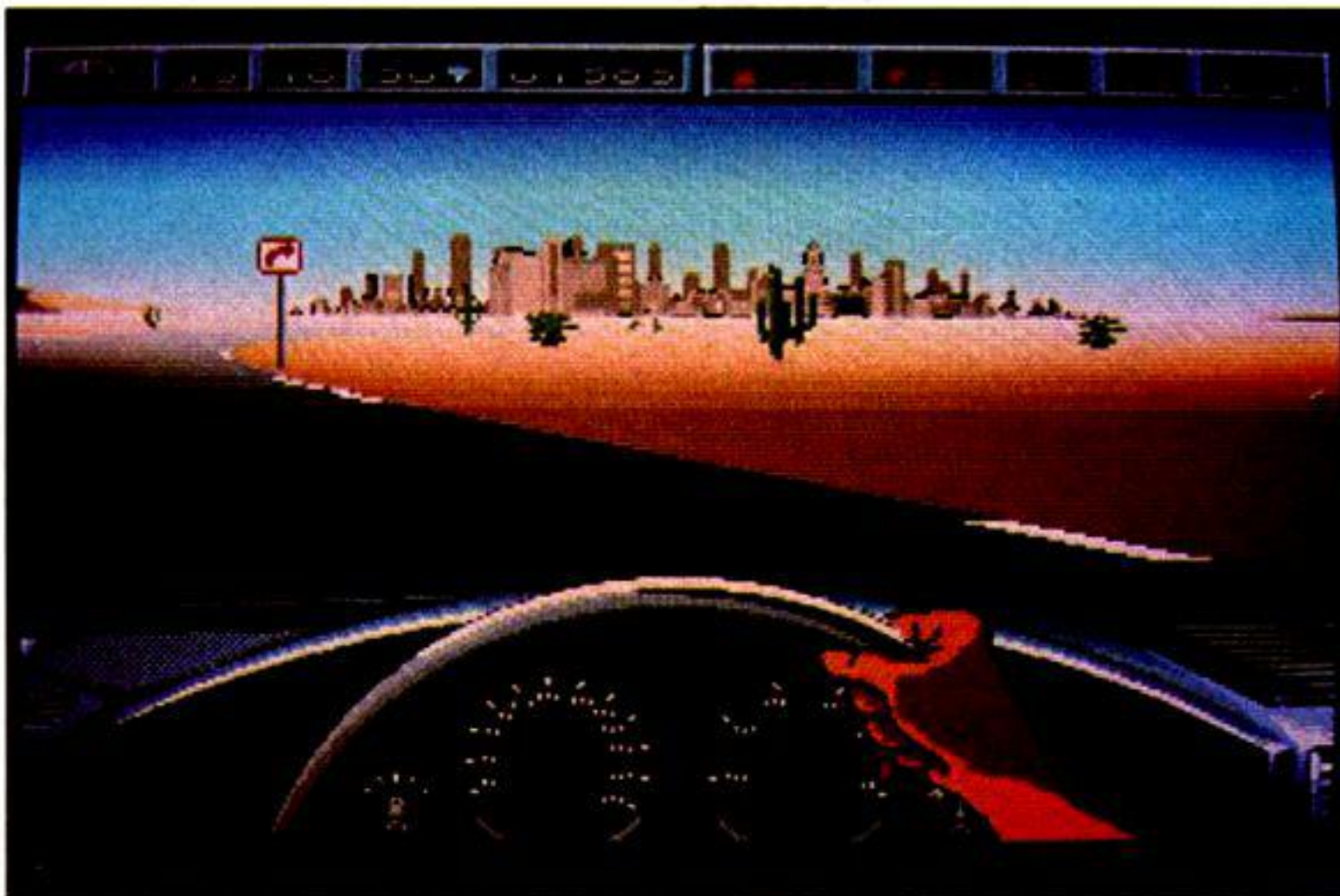
La tecnica

La grafica di scena è graziosa, con un orizzonte molto ben graduato (tracciato probabilmente in modo **Extra Halfbrite**) e le altre macchine sono dotate di moltissimi dettagli.

Le cose si complicano quando iniziano le animazioni, perchè la velocità di "aggiornamento" della strada e delle sagome dei veicoli non è così veloce come si vorrebbe: non vi sono più di sei fotogrammi tra l'auto alla massima grandezza e alla minima.

La musica di introduzione è stucchevole, ma gli effetti sonori sono molto buoni: le gomme stridono, il motore romba ed i proiettili esplodono veramente.

Il voto

Tecnica non ineccepibile, discreti incentivi. 6+

Se vi piace il genere horror, questo gioco fa proprio per voi: terrorizzatevi a volontà

Computer: Amiga inespanso
Comandi: Joystick, mouse
Tipo: Avventura animata
Softhouse: Ubisoft

ZOMBI

Questo videogioco è praticamente basato sul film *Day of the Dead* di George Romero, in particolare sulla scena del supermarket infestato dagli zombies. Tale film è stato il capostipite della serie sui "morti viventi", ovvero caratterizzati da trame studiate ad hoc per un pubblico che ama le trame infantili e le emozioni orrorifiche.

Il gioco

Si vestono i panni di 4 persone che, a bordo di un elicottero, atterrano proprio su una zona infestata dai resuscitati affamati di carne umana. La scena si svolge nei Grandi Magazzini, con molti piani da visitare, in cui si aggirano gli zombies che, normalmente, si fanno gli affari loro, ma quando rompiamo troppo si voltano e ci attaccano. Possiamo trovare armi varie, sparse per lo schermo, per abbatterli meglio, ma inizialmente si possono usare solo i pugni (eccitante, per chi è un perverso, lo spruzzo di sangue che zampilla dalla testa rotta dello Zombie). Dopo avere abbattuto uno zombie bisogna portarlo nei piani inferiori e metterlo in freezer, altrimenti resuscita nuovamente. Il compito è di recuperare carburante per scappare, facendosi anche strada tra i pochi umani sopravvissuti che ingaggiano una disperata lotta per la sopravvivenza.

La tecnica

Lo schermo è suddiviso in finestre: una è la vista tridimensionale della scena: i personaggi si muovono, raccolgono oggetti e menano gli zombie (o le prendono da loro).

Il movimento avviene cliccando su quattro icone a freccia, con funzioni differenziate a seconda della situazione. I quattro personaggi sono controllabili uno per volta, selezionando l'icona del protagonista interessato.

La grafica è molto buona, in particolare le sequenze di aggressione degli zombies ed i loro particolari orrorifici.

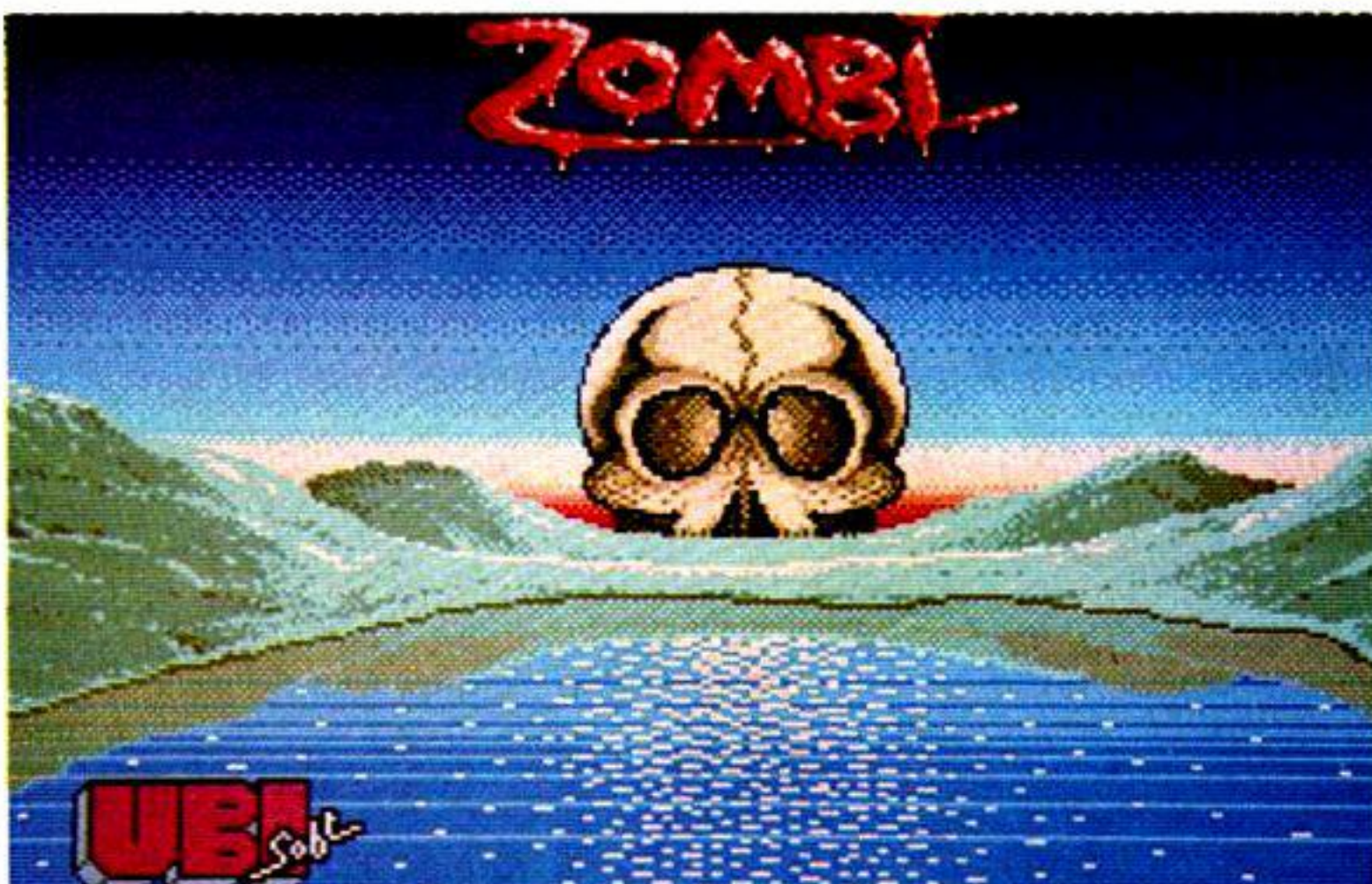
La colonna sonora, del famoso Davied Whittaker, è pertinente e crea un'atmosfera tesa, adatta alla trama di gioco.

Il voto

Bella grafica, buona tecnica, idea divertente anche se da horrormaniaci, leggermente ripetitivo. 8+.

CYBERBALL

Siete il manager di una squadra di androidi che giocano con una... palla...esplosiva!

Computer: Amiga inespanso
Tipo: Arcade
Gestione: Joystick
Tipo: Domark

Una variante estremamente futuristica del football americano, con droidi al posto dei concorrenti umani, convertita da un Coin op di grande successo della **Atari**.

Il gioco

Il gioco prevede sei squadre, due in torneo di allenamento e quattro in torneo professionisti.
Ciascun team contiene un gruppo di robots e si gioca rispettando piuttosto rigidamente le regole del football americano. Ad ogni turno una squadra gioca all'attacco, dovendo collocare il pallone oltre la linea di fondo della squadra avversaria in difesa; questa deve impedire agli avversari di raggiungere il loro scopo.

La prima importante differenza, rispetto allo standard, è che qui non vi sono limiti ai tentativi per fare avanzare il pallone di dieci yarde, ma se questo non sorpassa la meta o la linea centrale entro quattro tentativi, esplode. La qual cosa non solo inverte i ruoli di attaccanti e difensori delle squadre in campo, ma può anche arrecare seri danni ai robots. Ogni partita è suddivisa in **sei** periodi, ogni periodo dura **tre** minuti ed il conteggio non si arresta mentre si studiano, a video, gli schemi di gioco da praticare. Infatti, la parte di movimento è preceduta da una ovvia fase strategica, dove il giocatore deve scegliere gli schemi di movimento e di pasaggio di palla da far eseguire ai propri giocatori.

La tecnica

La grafica è da 3/4 dall'alto, abbastanza incisa, anche se le scelte dei colori avrebbero potuto essere migliori. In effetti assomiglia moltissimo a quella del Coin Op, e questo non può che essere un merito per quanto riguarda la fedeltà della conversione, se non per la pulizia della tecnica di scrittura del codice. Troviamo gradevoli effetti intermedi e alcuni brani di sintesi vocale.

Il voto

Idea originale, ben convertito. Chiede poco intelletto e molta grinta. 7-.

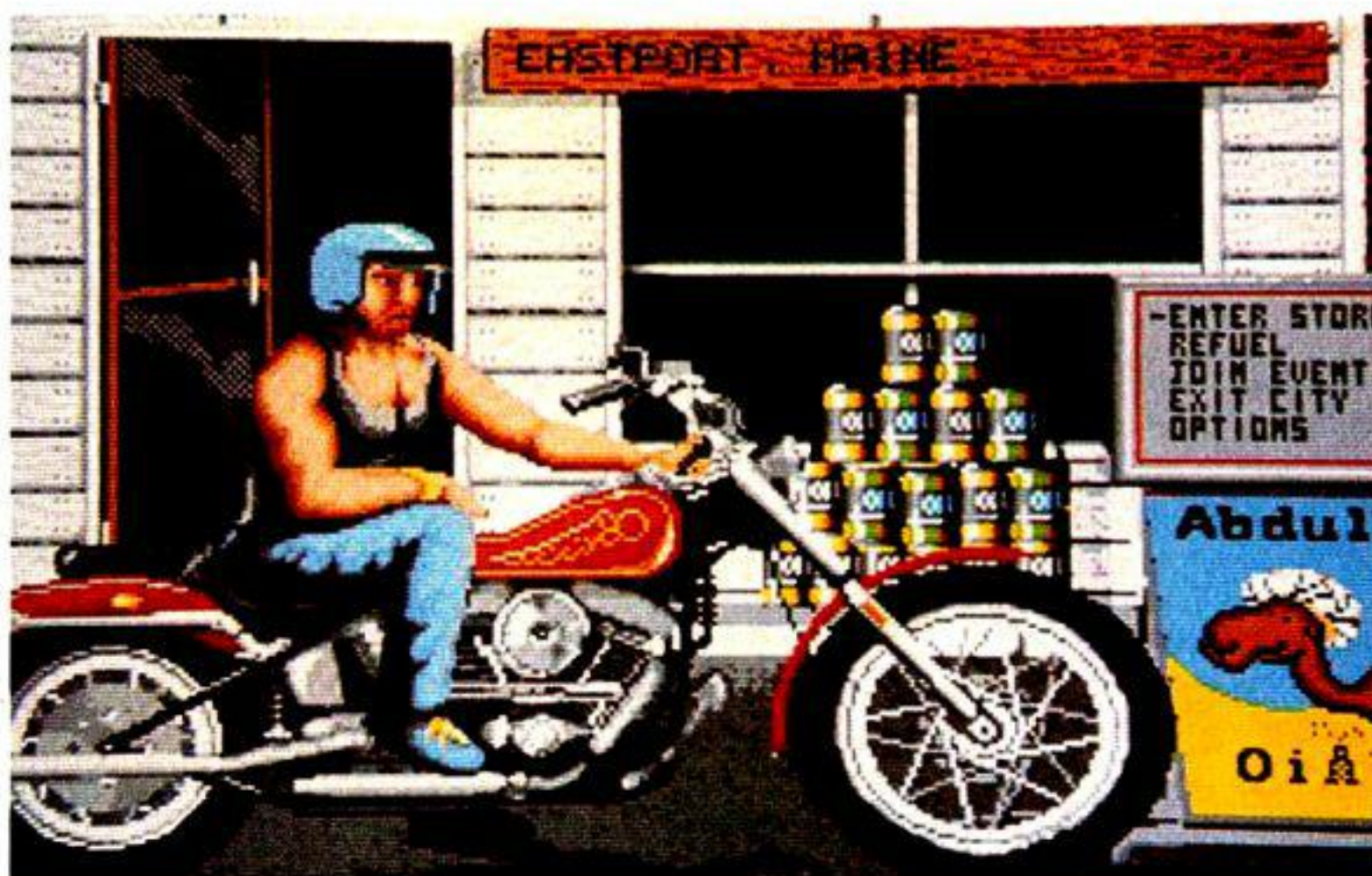
HARLEY DAVIDSON

Un corso accelerato per diventare impeccabili... teppisti in poche settimane

Computer: Amiga inespanso
Tipo: Simulazione moto
Gestione: Joystick
Softhouse: Mindscape

Lo scopo della vostra vita è parcheggiare la moto a pettine? Allora vorrete diventare il dio supremo della Harley Davidson!

Il gioco

E' un misto di azione arcade, consistente in corse in moto; si deve, però, anche scegliere l'equipaggiamento nella fase iniziale.
Lo scopo è di percorrere le strade sassose del **Maine** per raggiungere **Sturgis**, dove vi è la riunione dei fratelli a due ruote.

La simulazione della corsa non è difficile da padroneggiare, ed in breve tempo chiunque può imparare a guidare senza stirare i passanti od impastarsi frontalmente contro le macchine sull'altra carreggiata.

Più difficile, e lungo, è apprendere il senso delle fasi manageriali intermedie. Ad esempio, se ci si spupazza la proprietaria di una stazione di servizio (ma non vi sono scene animate, non è un gioco porno!) il prezzo degli articoli che compriamo (benzina, vettovaglie, birra, parti di ricambio) diventa solitamente più elevato. Forse abbiamo problemi di virilità?

Durante il gioco bisogna dimostrare di essere bravi col "mezzo", quello a due ruote; quindi ci si deve cimentare in varie prove, come ad esempio l'andatura lenta (evitando di ingolfare la moto) o l'accelerazione da scavezzacollo, oppure ancora il Poker Run (dove si devono raccogliere dalla moto le carte per fare una mano utile) e la scorribanda in qualche cittadella.

La tecnica

La grafica di movimento è buona, sebbene non sia certo particolarmente originale.
I fondali sono vari e abbastanza d'effetto. Le musiche, ovviamente, molto "metalliche" per accordarsi col tema del gioco, e gli effetti sonori sono più che sufficienti.

Il voto

Relativamente originale, moltissima varietà ma non moltissimo succo. 7-.

Siete incaricati di individuare molti oggetti, utili per salvare la Terra da una cometa

Computer: Amiga inespanso
Tipo: Avventura arcade
Gestione: Joystick
Softhouse: Novagen

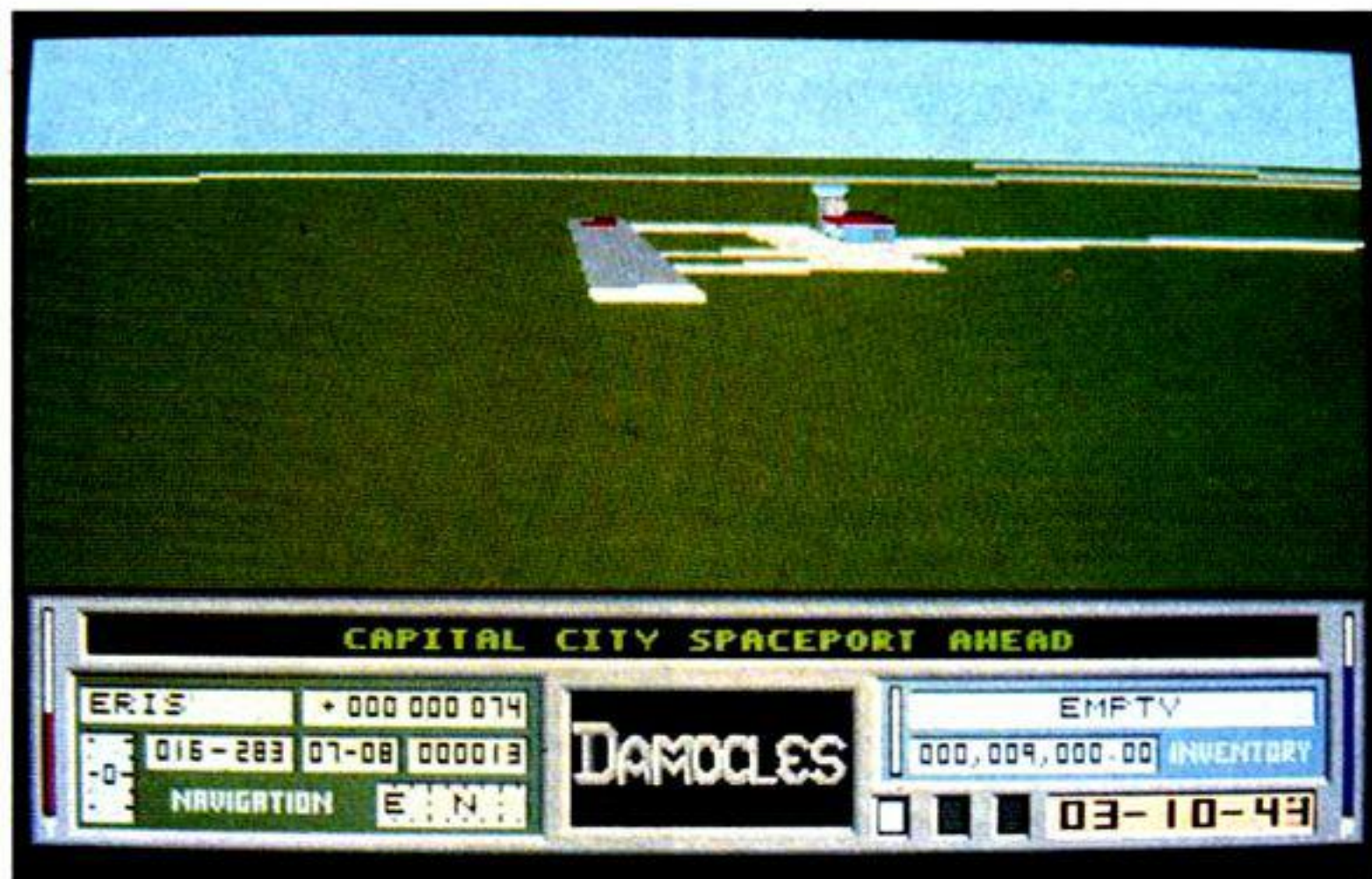
DAMOCLES

Un'enorme cometa sta per schiantarsi sul pianeta **Eris**. Se in **Mercenary** bisognava fuggire in tempo, qui bisogna evitare la tragedia in tre ore con l'aiuto del dottor **Benson** (un Personal Computer della nona generazione).

Il gioco

Si tratta proprio del proseguimento di Mercenary, visto che è consentito usare gli oggetti raccolti nel gioco precedente!
Il game presenta caratteristiche da "avventura", essenzialmente nel mettere a disposizione vari oggetti da sfruttare adeguatamente per "vincere" il programma.

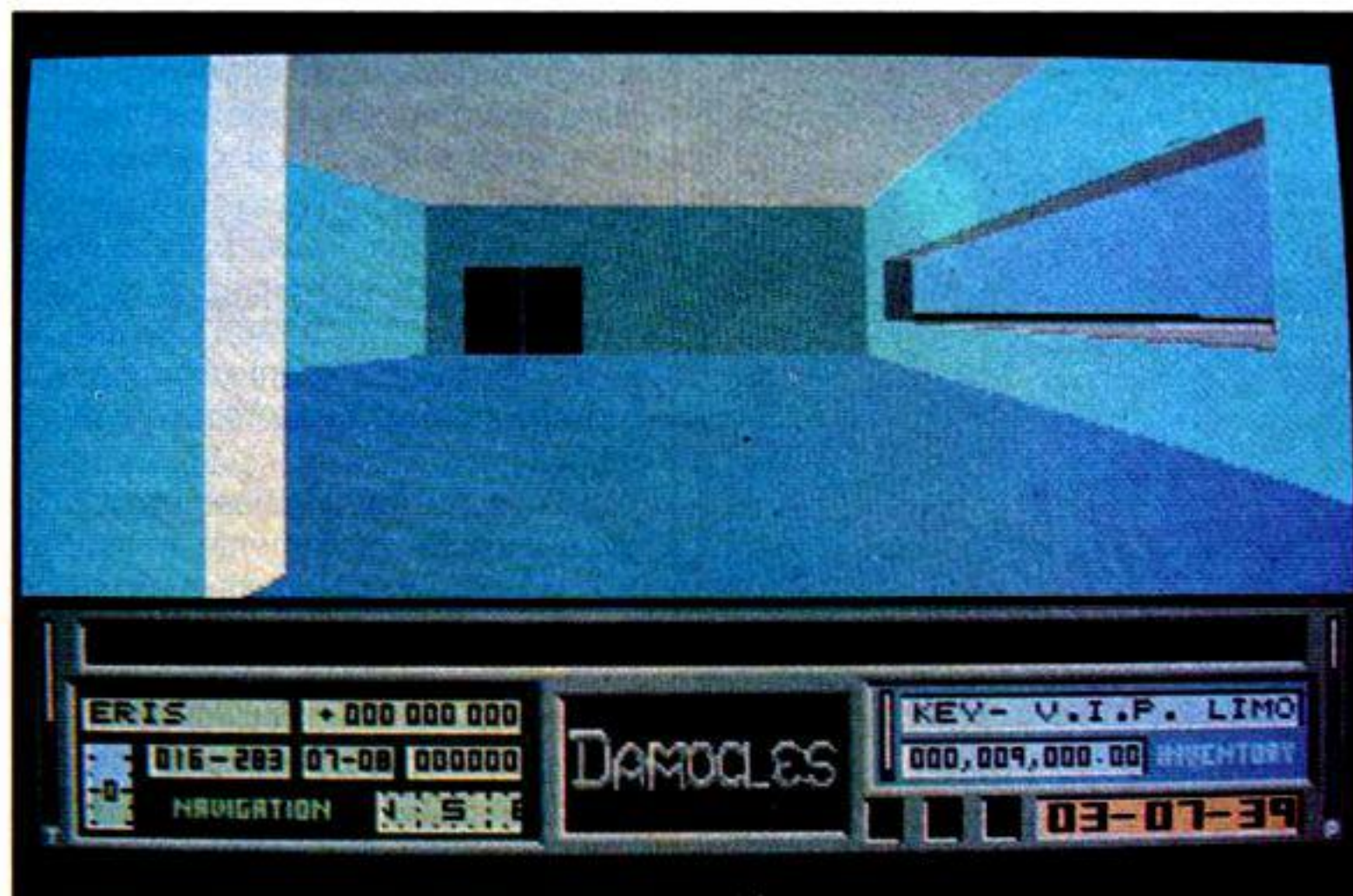
Tali oggetti sono particolarmente numerosi e disseminati su ben 9 pianeti e 18 lune.

Quando si afferra un oggetto, compare un menu che ne consente la manipolazione. Le funzioni che svolgono sono estremamente sofisticate; ad esempio vi è una videocamera che funziona proprio come una vera!

L'essenza del gioco, comunque, consiste, come in ogni buona avventura, nell'esplorazione di luoghi più o meno vari, con porte, finestre, corridoi ed ascensori (no, non è Impossible Mission!).

La tecnica

L'autore del programma è **Paul Woakes**, autore di Mercenary, ed ha migliorato notevolmente la tecnica di animazione grafica a vettori tridimensionali, che è infatti ora più incisa e fluida. La softhouse produttrice ha affermato che se si fosse curata di più la tecnica bruta necessaria ad un simulatore di volo, Damocles avrebbe potuto anche essere il più realistico e preciso programma di questo genere. L'audio è molto buono.

Il voto

Tecnica eccellente, interesse durevole nel tempo, di atmosfera. 9.

Un ennesimo gioco del tipo "spara e fuggi", ma con qualche novità di un certo rilievo

Computer: Amiga inespanso
Tipo: Scrolling zapper
Gestione: Joystick
Softhouse: Virgin

GEMINI WING

Un tipico programma "spara spara" a scorrimento verticale, con tanto di guardiani di fine livello, fedelmente ispirato all'omonimo Coin Op.

Il gioco

Uno o due giocatori, a turno, controllano una piccola astronave che si arrampica sullo schermo *zappando* con il laser, che fa sempre e soltanto fuoco verso il davanti, tutto ciò che incontra.

In ogni caso, gli alieni nostri avversari non sono soltanto volanti, ma troviamo anche vermi giganti che sparano, dalla bocca, sfere di fuoco come draghi (devono essersi montati la testa).

Non mancano vari oggetti da raccogliere sul percorso, come ad esempio vite extra e bonus punti, che a volte si materializzano dalle nubi di fuoco lasciati dagli alieni fatti esplodere.

Interessante è la **Gunball**, che funziona come una smart bomb, ma con effetti vari: a volte forma una Spirale Mortale che, ruotando, distrugge tutto ciò che incontra; altre volte esplode ortogonalmente. Alla fine di ogni livello troviamo l'immancabile guardiano, che è più grosso, pericoloso e meglio disegnato di tutti gli altri sprites.

Per ucciderlo bisogna portare dei colpi precisi e al momento giusto, dato che durante l'animazione che lo muove "copre" i propri punti vitali.

La tecnica

Sprites coloratissimi, animazioni fluide e precise.
Gli sfondi sono ben disegnati, anche se relativamente semplici come concezione, e risultano di effetto.

Gli effetti sonori sono forse anche migliori della grafica.

Il voto

Un gioco senza pretese, ma onesto nella sua semplicità. 6+.

Una guerra nucleare viene scatenata da strani personaggi: Khadaffi, Satcher...

Computer: Amiga inespanso
Tipo: Arcade strategico
Gestione: Mouse
Softhouse: US Gold

NUCLEAR WAR

Nonostante la guerra nucleare globale sia tutt'altro che piacevole, il programma presenta aspetti "divertenti".

Il gioco

Si vestono i panni di inquietanti personaggi, tra cui il colonnello Khadaffi, Mao the Pung, la signora Satcher, Infidel Castro o Ronnie Raygun. Tutti i comandi della stanza dei bottoni sono a vostra disposizione in un solo schermo. A seconda delle vostre decisioni appaiono altre schermate con animazioni divertenti e piacevoli. Lo scopo del gioco è di neutralizzare gli avversari; per farlo bisogna riscuotere i consensi della popolazione interna, costruire bombe nucleari, disporre le difese contro attacchi esterni, mandare in ricognizione aerei, far propaganda via radio, spernacchiare i nemici, eccetera.

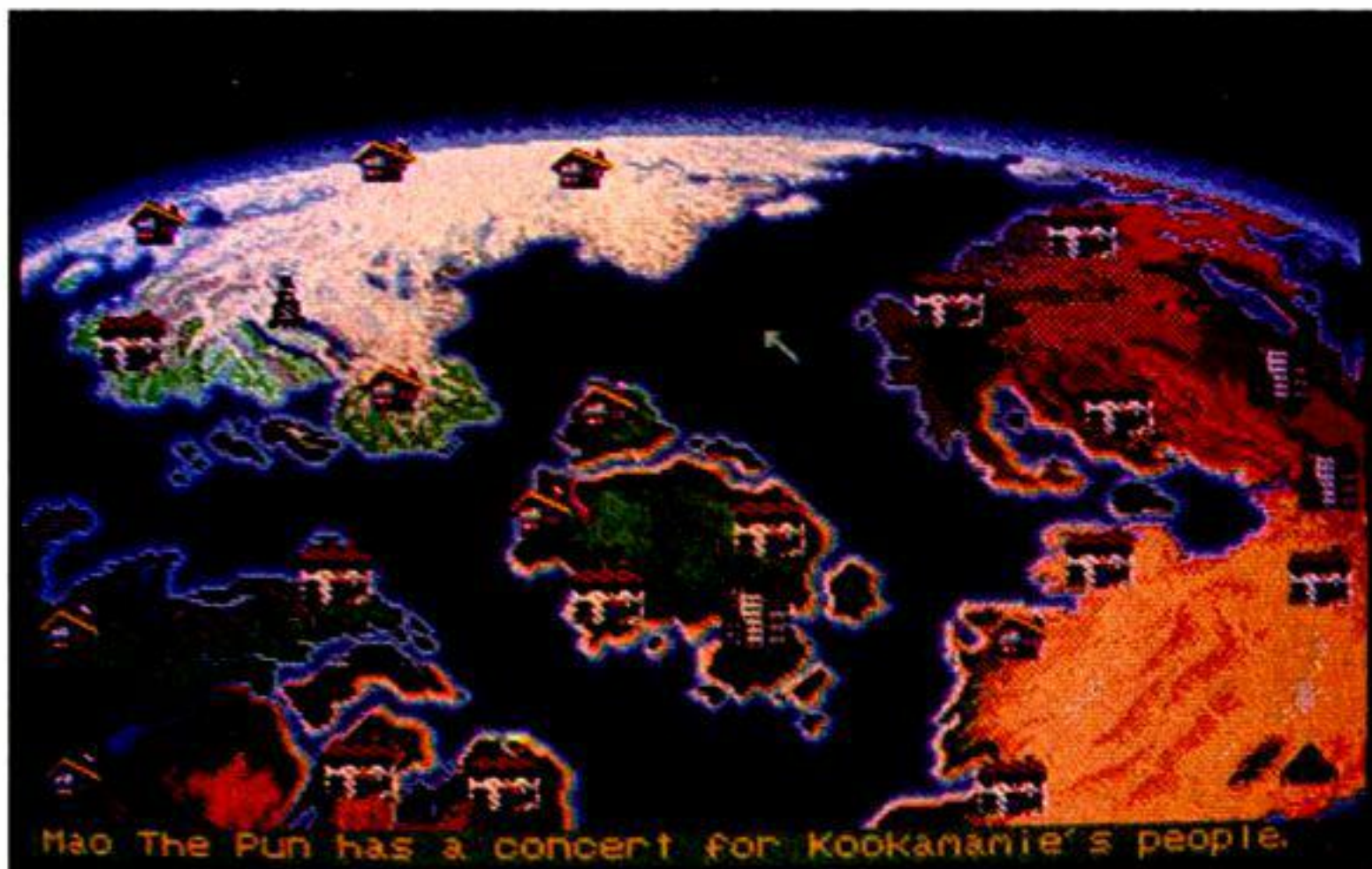
Oltre al semplice lancio di testate nucleari, dopo aver speso fondi per costruirle, è possibile risparmiare ed inviarle tramite il **Nuclear Postman**, un bombardiere che raggiunge una capacità distruttiva di cinquanta megaton, oppure con **Grim Reaper** che può trasportare in qualunque punto del globo un carico di cento megaton. Gli aeroplani hanno il vantaggio di portare più bombe, quindi di radere al suolo più di una città per volta. Altra forma di attacco, più subdola e da "guerra fredda", è la propaganda radio, che può indurre gli ascoltatori ad emigrare dal paese d'origine, con la conseguenza di indebolirlo. Il gioco ha caratteristiche strategiche notevoli, unite ad una grande dose di humor, proprio perchè simula tragedie eccessive. Ad ogni partita si ammazza mediamente un miliardo di persone, con i boss raffigurati in divertenti quadretti animati che ghignano e si scherniscono sullo stile di Spitting Image.

La tecnica

Le animazioni iniziali sono di eccellente qualità. La grafica del gioco è simpatica nelle caricature dei personaggi, ricchissima ed elegante nei gadget di selezione.

il realismo delle animazioni, tra cui la pressione del tasto di azionamento di una testata nucleare, è ammirevole. Gli effetti sonori sono ottimi.

Il Voto

Un gioco consigliato a tutti, da non perdere assolutamente. 9.

HAMMERFIST

Un gioco vario, anche se gli schemi sono quelli classici, arcinoti e, forse, stancanti

Computer: Amiga inespanso
Tipo: Arcade game
Gestione: Joystick
Softhouse: Activision

Il voto

Un buon gioco nel genere dei "salta", e spara. 7 1/2.

Dagli autori di **The Last Ninja**, Mev Dinch e Andrew Bond, un gioco senza samurai e molto futuristico.

Il gioco

Il **Maestro** è intrappolato nell'ultima di centoventotto schermate. Tanto per iniziare difficile, ogni schermata, o "stanza", è chiusa e si può accedere allo schermo successivo soltanto quando si sono distrutti oggetti, o nemici, in numero sufficiente.

Per proseguire non si deve soltanto distruggere il maggior numero possibile di alieni, robot, pesci e piante carnivore, ma anche raccogliere pezzi di computers.

Il nostro eroe è armato di un cannone laser in grado di trasformarsi spotaneamente nel suo alter ego femminile, **Metalisis**, quando serve più agilità e meno forza bruta.

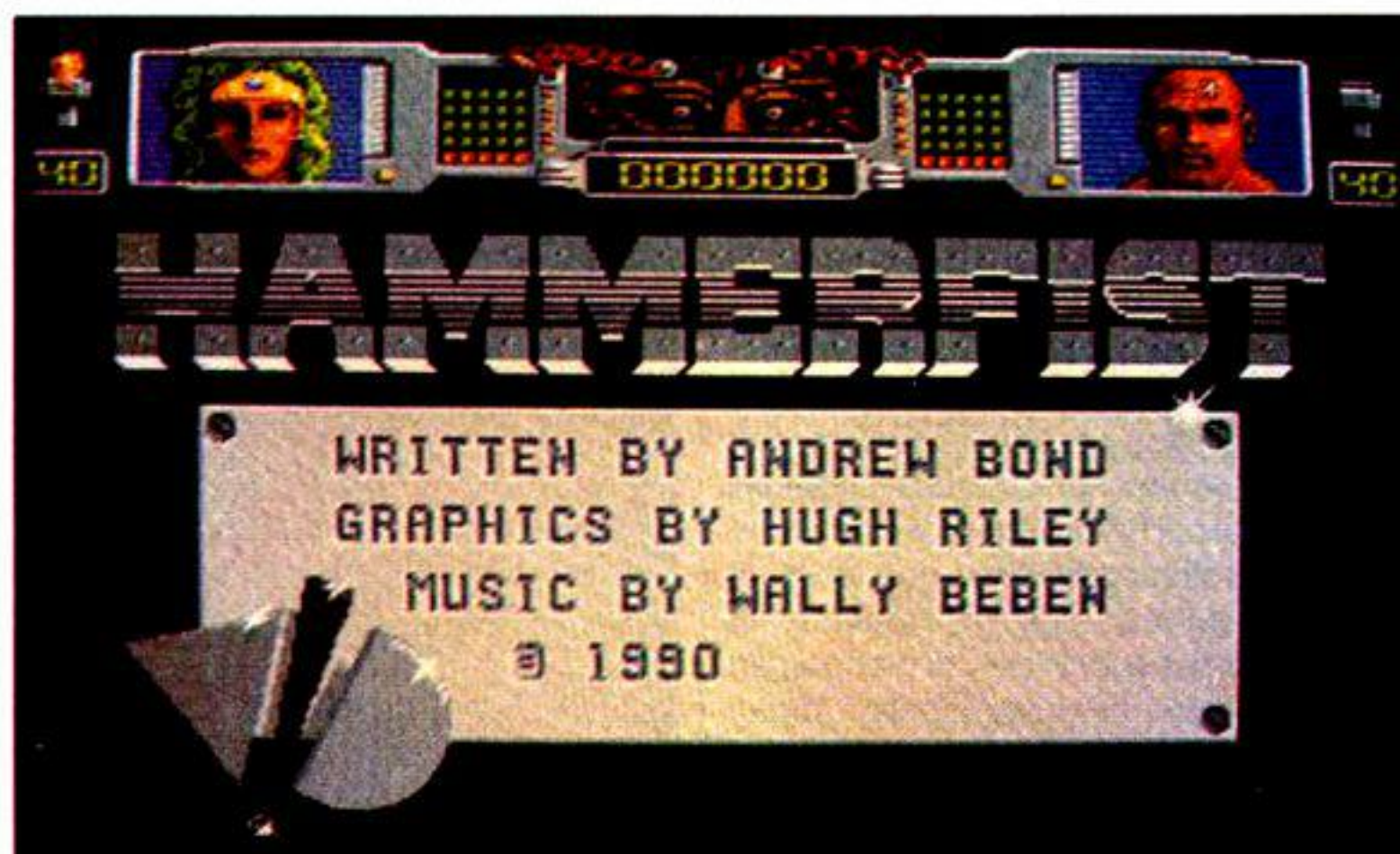
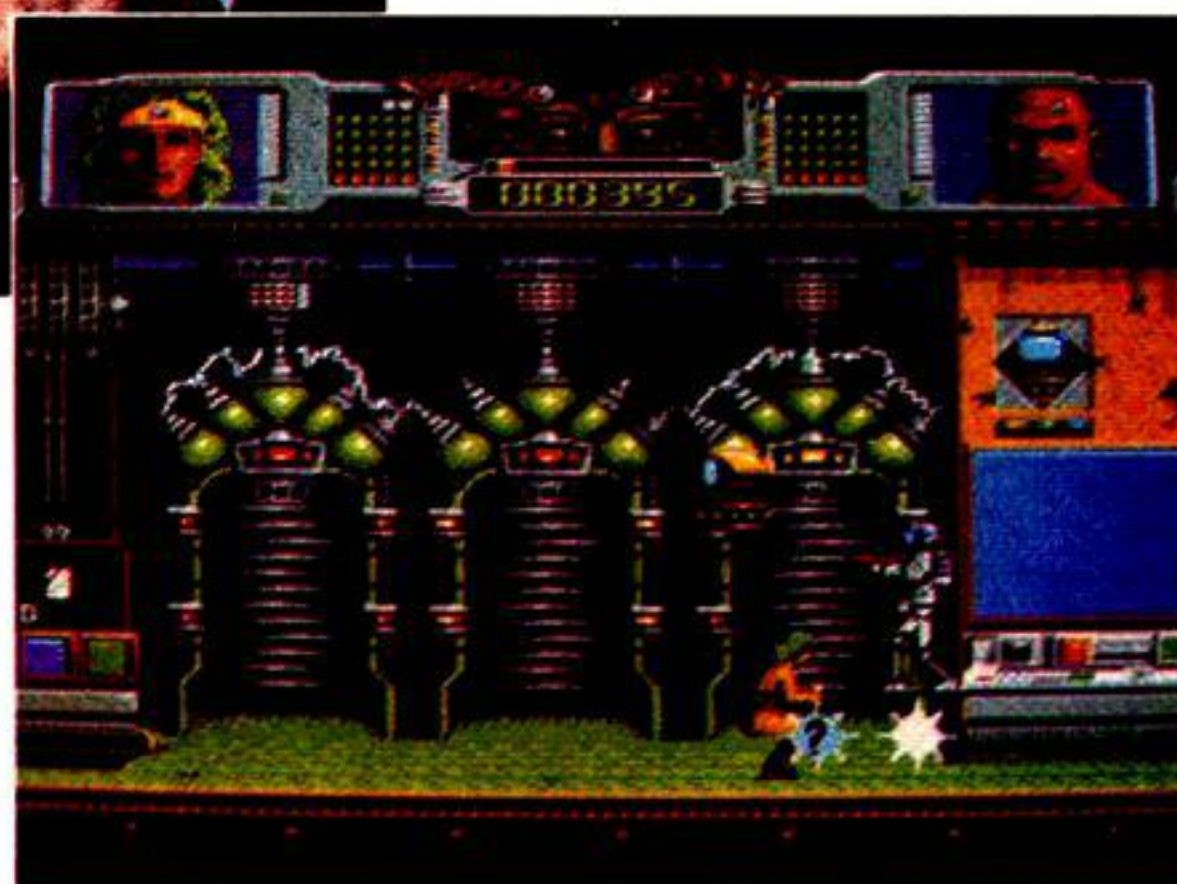
Durante il gioco bisogna raccogliere oggetti utili al prosieguo, ma non assomiglia proprio ad un'avventura!

La tecnica

Grafica molto varia nei centoventotto schermi, sebbene le animazioni dei personaggi risultino talvolta ruvide. La grafica di sfondo è relativamente curata e sempre diversa, almeno apparentemente. Gli effetti sonori sono all'altezza della situazione.

INFESTATION

La trama si ispira all'insuperabile film Alien: devi distruggere le uova dei mostri

Computer: Amiga inespanso
Tipo: Arcade spaziale
Gestione: Joystick, tastiera
Softhouse: Psygnosis

Un nome, **Psygnosis**, una garanzia. Sebbene non c'entri nulla con i **Barbarian** e **Shadow of the Beast**, si tratta di un programma tecnicamente e qualitativamente ottimo.

Il gioco

Dai tempi di **Aliens** sono usciti parecchi programmi basati su stormi di alieni che usano uova per riprodursi. In effetti Infestation si ispira al film, ma solo nelle basi della trama, trasposta nella ricerca della stazione **Alpha II** sulla luna di Xelos.

Le più recenti trasmissioni di quella stazione sono diventate incomprensibili, e pare che la causa sia dovuta ad insettoni che nascono da uova disseminate un po' ovunque.

Il gioco inizia con l'eroe, **Kal**, che viene inviato su Xelos (in una bellissima animazione) armato di cartucce **Cyanide** con le quali può distruggere le uova, una unità di pressurizzazione, una unità di manovra (MMU), che gli consente di spostarsi sulla superficie del pianeta, ed un cannoncino per distruggere i droidi controllati dagli avversari.

Lo scopo è di trovare le uova per scaricarvi vicino le cartucce esplosive di Cyanide, facendosi largo tra droidi, corridoi senza luce, aree contaminate e tante altre avversità.

La salvezza dell'universo è nelle vostre mani!

La tecnica

La costruzione di poligoni tridimensionali in tempo reale per simulare il movimento, visto dall'interno dell'unità semovente, non è affatto una tecnica nuova, ma è comunque di grande effetto e ben realizzata.

Infestation abbina una consolle di manovra con grafica in bitmap standard a quella 3D a poligoni senza flickering tramite il modo Amiga, noto come **Dual Playfield**.

L'attenzione per i dettagli è superba. Le introduzioni e le schermate introduttive sono ai massimi livelli attuali per Amiga.

La musica è molto buona, così come gli effetti sonori propri del gioco. Si pensi che alla Psygnosis hanno assunto sei artisti specializzati in computer graphics solo per realizzare sprites e grafiche di sfondo e decorative!

Il voto

Non può che essere un best seller. 8/9.

Potete giocare addirittura in quattro e divertirvi a guidare altrettante automobili

Computer: Amiga inespanso
Tipo: Arcade
Gestione: Joystick
Softhouse: Activision

HOT ROD

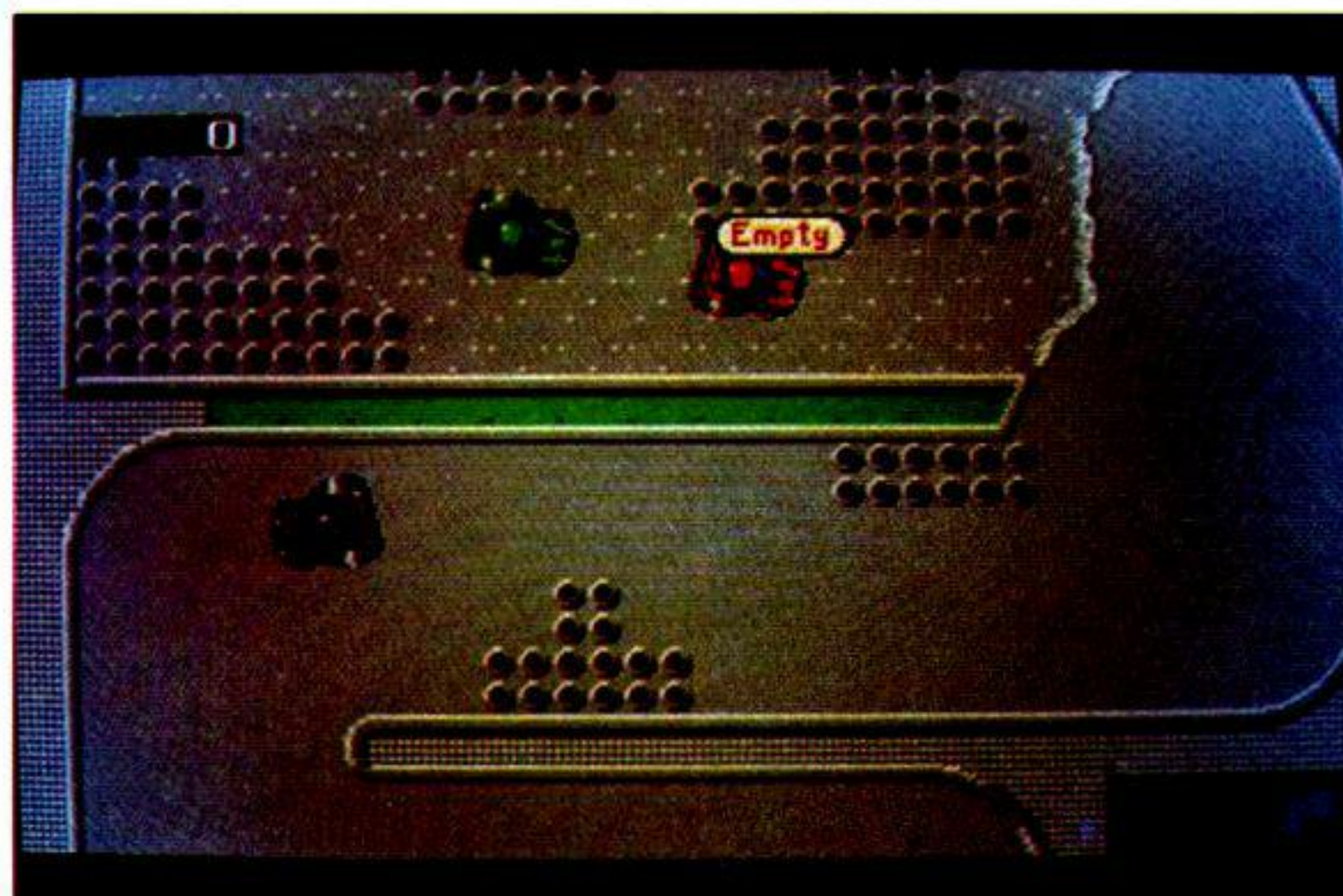
Convertito per Amiga, da un famoso Coin-op, largamente basato sul SuperSpring di Atari, questo videogame combina il meglio dei concetti di macchine veloci, piste tortuose, accessori per macchine originali e quattro giocatori contemporaneamente all'opera.

Il gioco

Le automobili corrono lungo strade semplici, strutturalmente, ma di difficile percorrenza a causa del modo in cui si guidano: le vetture sono viste quasi perpendicolarmente dall'alto.

Sul percorso si trovano anche icone di bonus, che forniscono punti extra oppure carburante supplementare, che può altrimenti terminare prima dell'arrivo.

Le macchine non vengono mai lasciate indietro: se qualcuno va fuori strada o ha un incidente, viene automaticamente "spinto", è la parola più giusta, in avanti affinchè non scompaia dal riquadro della pista visualizzata, che segue fedelmente la prima macchina.

Le scene sono numerosissime, sempre diverse, con particolari grafici di sottofondo a volte divertenti.

La tecnica

Il controllo è semplicissimo: il pulsante di fuoco accelera, con la manopola si fa ruotare in senso orario, o antiorario, il muso del veicolo. Uno dei punti di forza del videogioco è proprio la semplicità.

La grafica è gradevole, molto veloce, anche se non presenta alcuna particolare ricercatezza tecnica. Gli effetti sonori sono dello stesso livello, e con poco sforzo si sarebbe probabilmente potuto fare molto di più. Interessante è la possibilità di usare un **Dongle** in porta seriale, ovvero uno di quei **joystick doppi**, molto in voga qualche tempo fa, che consente a due avversari di usare il joy, ed eventualmente ad altri due di usare la tastiera.

Il voto

Un gioco simpatico, adatto per le compagnie di videogamers. 7.

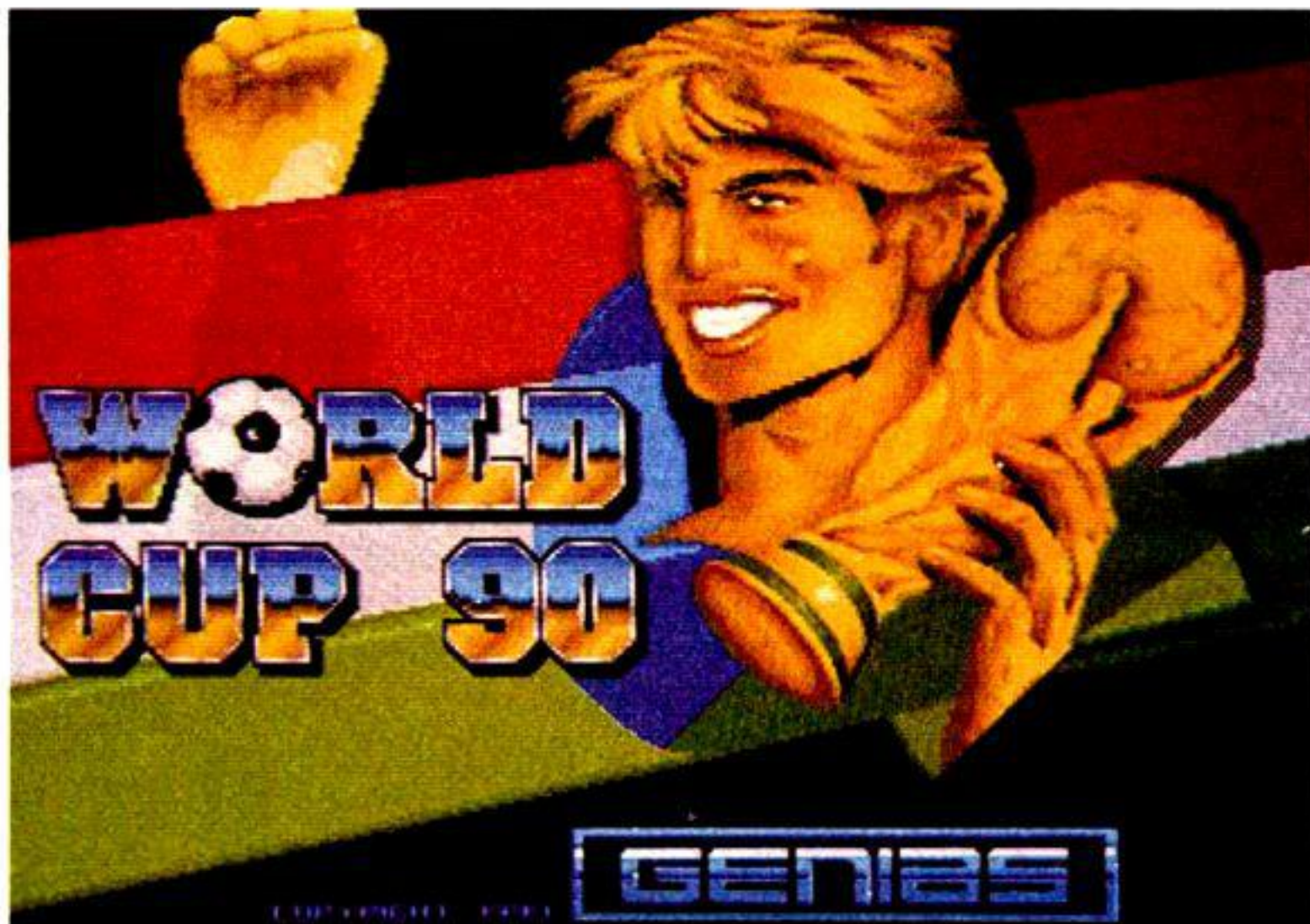
Una nuova simulazione calcistica, che presenta pochi vantaggi r ispetto al più noto " Kick Off"

Computer: Amiga inesp. e C/64
Tipo: Calcio
Comandi: Joystick
Softhouse: Genias

WORLD CUP 90

Sviluppato da un team di programmatori italiani, è un classico simulatore di partite di calcio sulla scia degli altrettanto classici "pseudo tridimensionali", cioè con grafica laterale dall'alto.

Il gioco

E' possibile disputare semplici match o un intero campionato, scegliendo addirittura le magliette delle varie squadre. Vi sono parecchie opzioni di gioco, anche se non certo in numero faraonico come in Kick Off, tra le quali la durata materiale del tempo e le caratteristiche del terreno. Tutti i controlli "gestionali", non inerenti il gioco puro, sono effettuati su schermi con icone molto belle.

I giocatori corrono per il campo in modo realistico, anche se hanno sempre il vizio di fare rapidi dietrofront senza spiegazione mentre siamo noi che portiamo avanti la palla.

La tecnica

Buona la grafica, tra le migliori nel suo genere prospettico.Le sequenze di animazione sono buone e non risentono del numero di giocatori visualizzati. Il portiere si tuffa bene (quasi meglio di Zenga).Con un megabyte di RAM si ha a disposizione anche la **moviola** per il replay (per fortuna non hanno messo anche Biscardi!).

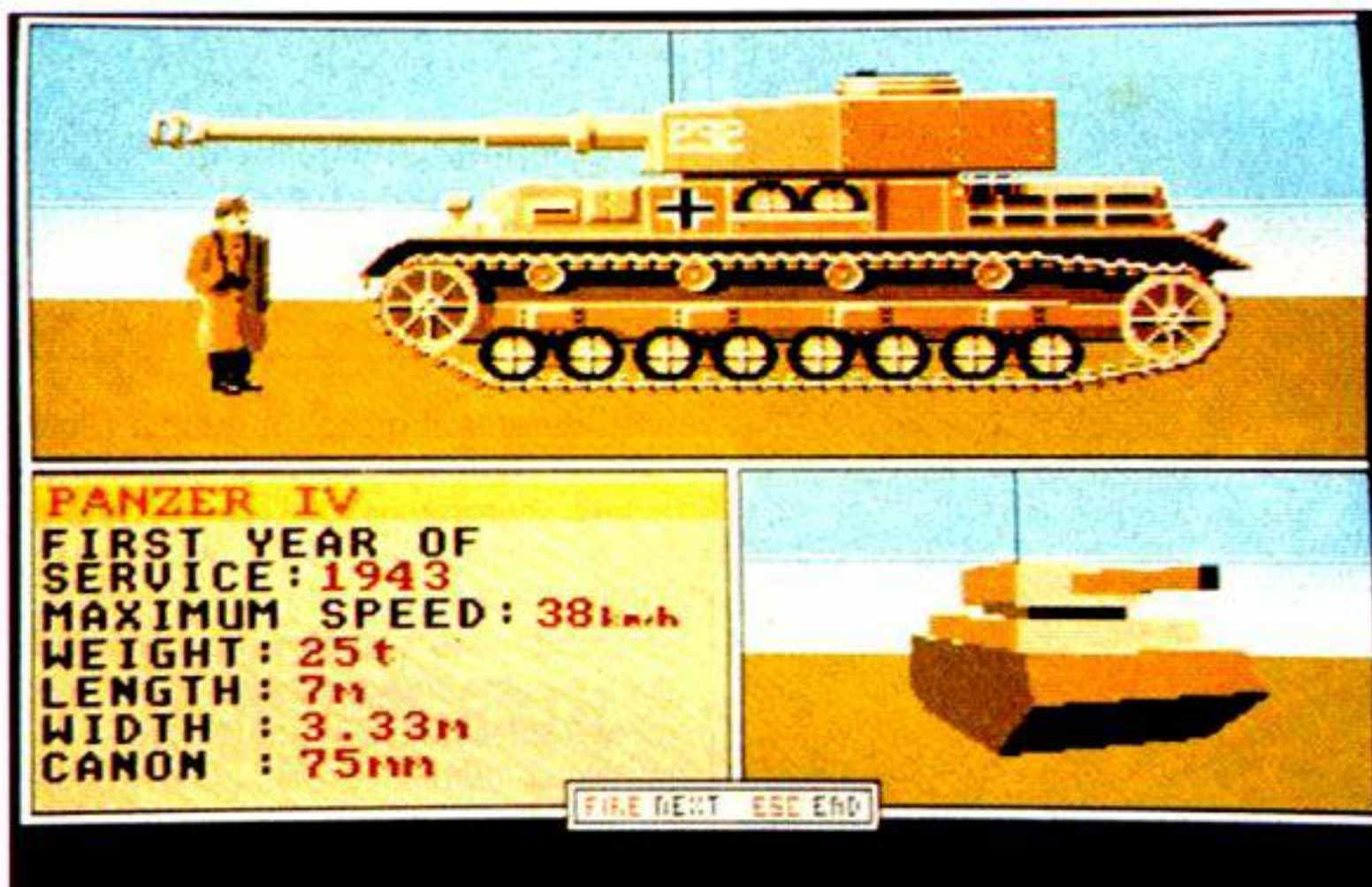
Tramite una apposita interfaccia, acquistabile a parte, si può giocare anche in quattro contemporaneamente.

Il voto

Un bel gioco, tra i migliori di un genere trito e ritrito. 9--.

SHERMAN M4

Gli appassionati guerrafondai pilotano un carrarmato in varie azioni belliche

Computer: Amiga inespanso
Tipo: Arcade game
Gestione: Joystick
Softhouse: Loriciel

Dopo **Tank Platoon** e **Conqueror**, ecco un'altra simulazione di guerra "a bordo" di un cartone animato, con tanto di grafica vettoriale.

Il gioco

Lo **Sherman M4** fu un carro armato molto usato durante la seconda guerra mondiale. Noi lo possiamo pilotare, a scelta, in campagne di guerra ambientate in Normandia, Ardenne o Nord Africa.

Ogni campagna militare, condotta da una pattuglia formata da quattro carrarmati e due jeep di appoggio, è divisible in più fasi, o missioni, di difficoltà ovviamente progressiva.

In effetti, è consigliabile inizialmente compiere ciascuna missione separatamente.

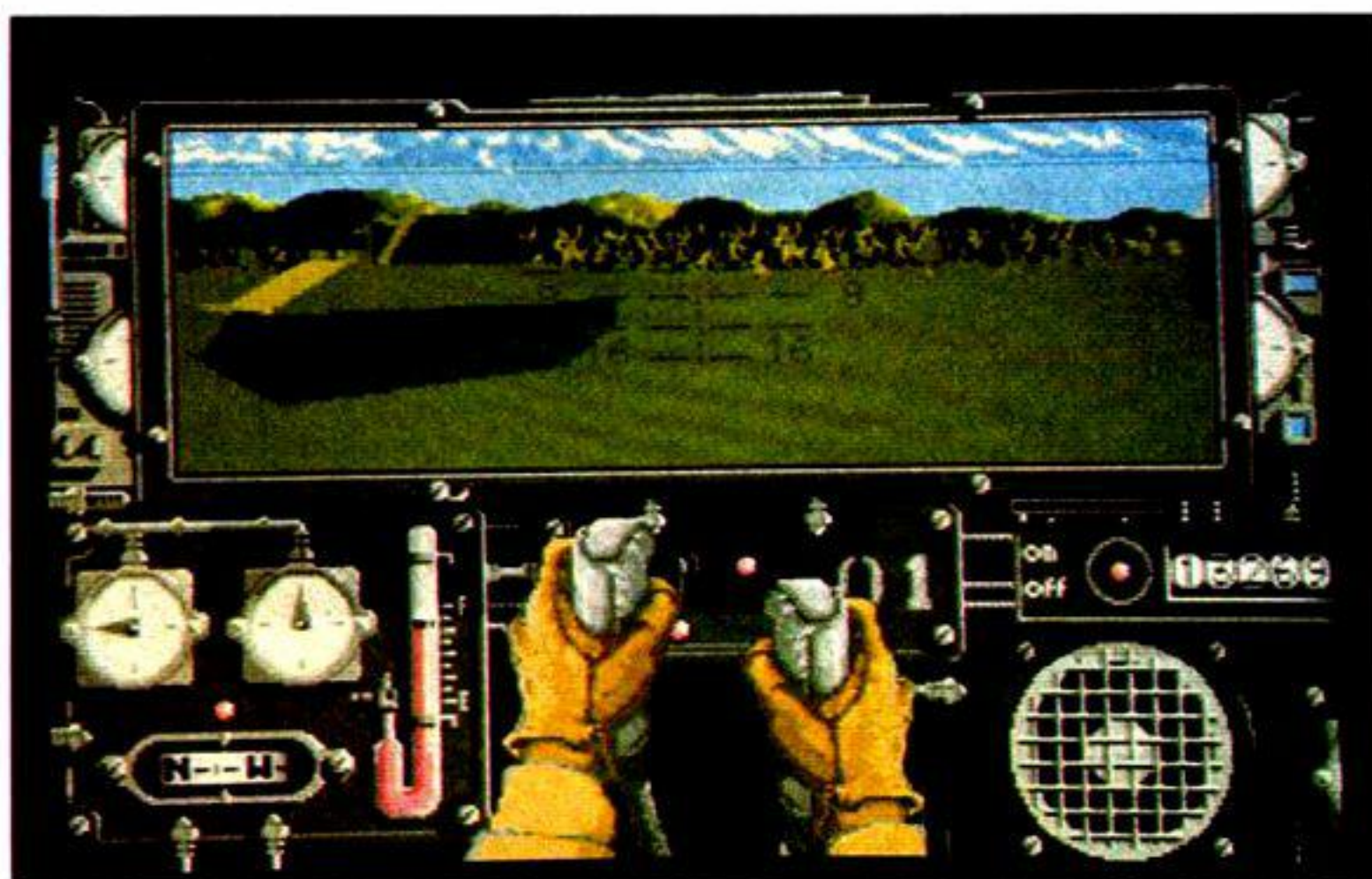
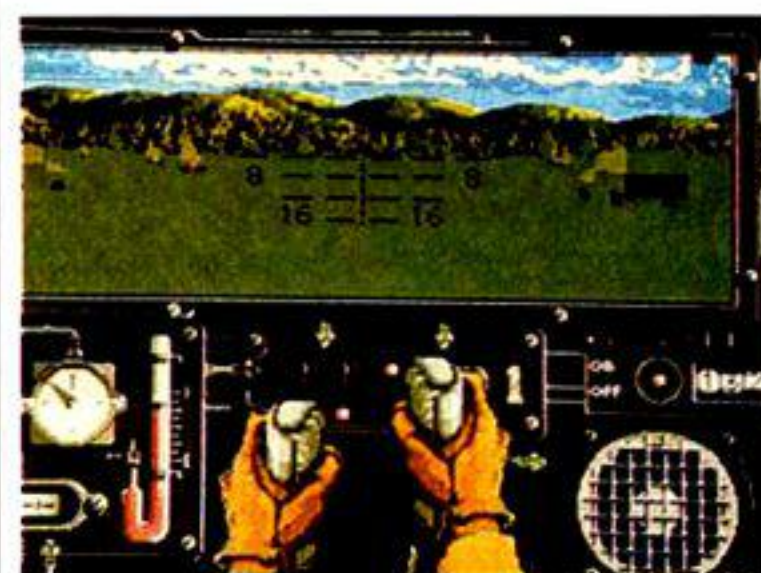
Il gioco non è un semplice arcade a grafica vettoriale, ma risulta quasi esclusivamente "strategico", tenendo sullo schermo la mappa ed impartendo per radio i comandi ai mezzi della truppa. Le comunicazioni tra i soldati ed il quartier generale avviene tramite una finestra posta nella porzione inferiore dello schermo.

La Tecnica

La grafica è molto buona, veramente ai massimi livelli per quanto riguarda la vettorialità e la velocità di calcolo.

E' possibile scegliere tra la vista dall'interno del carrarmato, dall'esterno (quattro direzioni) oppure dal telescopio, che svolge la funzione di canocchiale. Un punto notevolmente scarso è rappresentato dagli effetti sonori, certamente non all'altezza della grafica. L'interfaccia utente è piuttosto semplice ma molto efficiente.

Il voto

Ottima tecnica, a parte i suoni. Scarsi gli incentivi. 7 1/2.

DYNASTY WARS

La vendetta di quattro Samurai che hanno subito un gravissimo torto

Computer: Amiga inespanso
Tipo: Arcade
Gestione: Joystick
Softhouse: US Gold

Sebbene i protagonisti siano **Samurai**, non è un gioco tipo International Karate o The Way of Exploding Fist, pur se rimane una lotta all'ultimo sangue.

Il gioco

Cina, anno 184 dopo Cristo. La famiglia **Han** deve affrontare una lotta a scrolling orizzontale contro alcuni generali, rei di aver distrutto i loro possedimenti. In un altro tipo di gioco, costoro si chiamerebbero "guardiani di fine livello", visto che li si incontra dopo avere attraversato orizzontalmente tutti gli schermi che compongono ciascuna delle fasi di gioco, uccidendo tutti i mascalzoni che si incontrano a colpi di sfere di fuoco e spada.

I guerrieri Han sono quattro, ciascuno dotato di una differente abilità e tipo di armamento. Se per i generali occorre solitamente un colpo molto forte, inferto da un'arma difficile da portare, per affrontare le torme di nemici che saltano come cavallette con occhi a mandorla occorrono combattenti più agili.

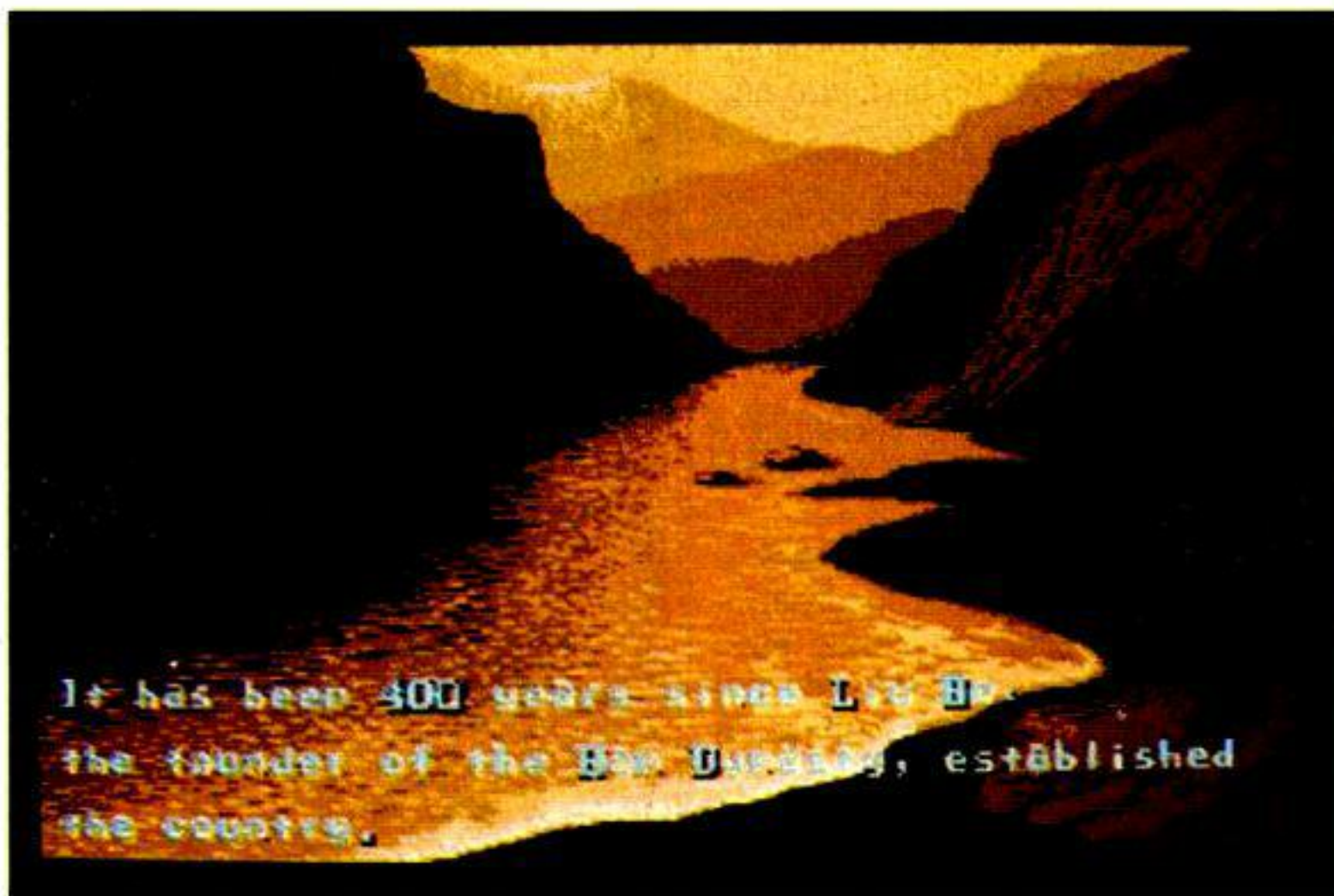
Purtroppo gli incentivi non sono molti: due modi di attacco e torme di nemici da colpire, questo è tutto. In ogni caso si ha un senso di tensione persistente mentre si gioca e la grafica contribuisce a dare un certo "charme" ad un programma altrimenti banale.

La tecnica

Le schermate introduttive, lunghe da trascorrere, creano un'atmosfera di leggenda. I personaggi descritti saranno rivisti durante il gioco, in quanto i loro volti sono riprodotti nella sezione inferiore dello schermo per illustrare lo stato di salute, in base alle smorfie di dolore ed alle ferite sul viso.

Gli sfondi sono eccellenti, ma l'area di gioco sembra troppo piccola per fornire dettagli sufficienti. Quando lo schermo si riempie di nemici non si capisce quasi più nulla! La musica è di atmosfera, ma noiosa, anche perchè si ripete a cicli relativamente brevi e non è interrotta da alcun effetto sonoro durante le lotte.

Il voto

Idea standard, tecnica non perfetta, ma ottima atmosfera. 6 1/2.

```
/*
  OPERAZ.C
  Esegue la somma, la sottrazione, il prodotto ed il quoziente tra 2 numeri
  By Mariani Giancarlo 1990
  Linguaggio : Turbo C 2.0
*/
void main()
{
  float V1,V2;
  int Scelta;
  char ch;

  do
  {
    clrscr();
    printf("Introduci due numeri separati da una virgola e premi CR\n");
    scanf("%f,%f",&V1,&V2);
    gotoxy(1,5);
    printf("1 - Somma\n");
    printf("2 - Sottrazione\n");
    printf("3 - Prodotto\n");
    printf("4 - Quoziente\n\n");
    printf("Scegli : ");
    do
    {
      gotoxy(10,10);
      scanf("%d",&Scelta);
      if ((Scelta<1) || (Scelta>4)) printf("\nScelta errata\n");
    }
    while ((Scelta<1) || (Scelta>4));
              gotoxy(1,12);
              if (Scelta==1) printf("La somma vale : %f\n\n",V1+V2);
              if (Scelta==2) printf("La sottrazione vale : %f\n\n",V1-V2);
              if (Scelta==3) printf("Il prodotto vale : %f\n\n",V1*V2);
              if ((Scelta==4) && (V2!=0)) printf("Il quoziente vale : %f\n\n",V1/V2);
              if ((Scelta==4) && (V2==0)) printf("Divisione per zero!!\n\n");
    printf("Premi un tasto ('q' per finire) ");
    ch = getch();
  }
  while (ch != 'q');
  printf("\n\nfine\n");
```

Il listato scritto in Turbo C Borland

spondenti istruzioni del C, per cui potete riferirvi alle versioni Basic.

Ch=getch()

L'istruzione **ch = getch()** aspetta la pressione di un tasto e ne pone il valore nella variabile **ch**.

L'istruzione equivalente del Basic è **Inkey$**. La successiva **while** chiude il ciclo **do** aperto prima, e

fa ripetere il programma fino alla pressione del tasto **q** (while ch != 'q'); si noti la presenza dell'apice (') e non delle virgolette (") per la definizione di un carattere stringa.

In Basic, il ciclo viene chiuso da **Wend**.

L'ultima parentesi graffa chiusa (}) indica il termine del programma **main**.

Turbo C Borland

Il programmino di esempio è stato scritto utilizzando il compilatore Turbo C (versione 2.0) della Borland, il potentissimo linguaggio che gira su compatibili Ms-Dos. Turbo C unisce la potenza di un linguaggio compilatore alla semplicità di scrittura e di esecuzione di un linguaggio interprete. Turbo C possiede al suo interno:

- Un completo e versatile **editor**, che consente la scrittura del programma sorgente.
- Un **compilatore** ed un **linker**, che consentono di compilare e quindi di far girare il programma.
- Un **segnalatore** di errori con help in linea; ciò significa che, in presenza di un qualsiasi errore (o dubbio) è sufficiente premere un tasto per avere, su video, una completa descrizione di ciò che è successo.
- Un **help** in linea su tutte le parole chiave, che permette di esaminare, all'istante, la sintassi o l'uso di una determinata istruzione del C.
- Un potente **debugger**, che permette di eseguire il programma passo - passo, di controllare i valori delle variabili, di inserire breakpoint, ed **altre funzioni** (troppe, per descriverle tutte)che semplificano notevolmente la ricerca degli errori logici di programmazione.

Tantissime altre caratteristiche (gestione di tutte le schede grafiche commercializzate e di quelle seriali, gestione diretta della memoria) e soprattutto un prezzo più che accessibile, ne fanno un prodotto unico nel suo genere.

Inutile dire che per affrontare **seriamente** il linguaggio C è indispensabile riferirsi costantemente al manuale di istruzioni che, voluminoso e completo, non può essere considerato un "accessorio" del linguaggio: il C non è un videogame intuitivo...

Ne consegue la necessità inderogabile di procurarsi la versione originale del **Turbo C Borland** che, come già detto, viene offerta ad un prezzo che sicuramente compensa le enormi soddisfazioni ottenibili dall'uso di un moderno compilatore.

Come scrivere in C

Supponendo che abbiate già installato correttamente (come descritto sul manuale) **Turbo C** sul vostro calcolatore, la sequenza di operazioni da effettuare, per scrivere e veder girare il programma, è la seguente:

1) Richiamare il Turbo C da Ms-Dos tramite il comando **TC**. Verrà visualizzata una schermata di presentazione con il copyright e la versione del Turbo c.

2) Premere i tasti **Alt + F** per far apparire la finestra **file**, in alto a sinistra dello schermo, che descrive le operazioni possibili.

3) Selezionare con i tasti **freccia** in alto ed in basso l'operazione **load**; quindi premere Enter. I punti 2 e 3 possono essere sostituiti dalla semplice pressione del tasto **F3**.

4) A questo punto Turbo C richiede il nome del programma da caricare.

Digitate **Operaz.c** e premete Enter. Se il file è presente su disco (ma non dovrebbe...) verrà caricato, altrimenti quel nome verrà considerato come nome di un nuovo file. Se premete qualche tasto errato, **Esc** permetterà di uscire da qualsiasi situazione, riportando il Turbo C, un passo alla volta, alla schermata iniziale.

5) A questo punto siete in **edit**, e potete scrivere il programma sorgente.

L'uso dell'editor è abbastanza semplice ed intuitivo: con i tasti cursore, compresi **PgUp**, **PgDn**, **Home**, **End**, ci si sposta in lungo ed il largo per il programma; i tasti **Ins**, **Del**, **Bs**, inseriscono o cancellano caratteri. I comandi per la gestione dei blocchi e per la ricerca di testi sono identici a quelli di **WordStar**. E' bene tener presente alcune fondamentali considerazioni.

Il C differenzia le lettere **minuscole** dalle **maiuscole**, per cui bisogna fare attenzione a copiare il programma esattamente come è riportato sulla rivista. In particolare, le **parole chiave** del C andranno scritte interamente in minuscolo.

Non bisogna dimenticare, inoltre, i caratteri di punto e virgola (**;**) posti alla fine di ciascuna riga.

6) Una volta scritto il programma, è bene registrarlo subito, per evitare accidentali perdite dovute ad un (improbabile) inchiodamento del sistema.

La sequenza per registrare il programma è la seguente: premere i tasti **Alt + F**; selezionare, tramite cursore, l'operazione **Save** e premere Enter.

L'intera sequenza può essere sostituita dalla pressione del solo tasto **F2**.

7) Per compilare il programma bisogna premere i tasti **Alt + R** e, subito dopo, selezionare l'operazione **Run**.

8) A questo punto Turbo C inizia a compilare il programma sorgente appena introdotto.

Se non vi sono errori, subito dopo la compilazione il programma viene fatto girare e potremo controllarne la corretta esecuzione. L'elaborazione può essere interrotta in qualsiasi momento tramite la pressione dei tasti **Ctrl + Scroll Lock** (Break).

Se, invece, il compilatore riscontra errori, la maschera di compilazione su video ci informa sulla loro

```
'
' OPERAZ.BAS
' Esegue la somma, la sottrazione, il prodotto ed il quoziente tra 2 numeri
' By Mariani Giancarlo 1990
' Linguaggio : QuickBasic 4.5
'
  DIM V1, V2 AS SINGLE
  DIM Scelta AS INTEGER
  DIM Ch AS STRING

  DO
    CLS
    PRINT "Introduci due numeri separati da una virgola e premi CR"
    INPUT V1, V2
    LOCATE 5, 1
    PRINT "1 - Somma"
    PRINT "2 - Sottrazione"
    PRINT "3 - Prodotto"
    PRINT "4 - Quoziente": PRINT
    PRINT "Scegli : "
    DO
      LOCATE 10, 10
      INPUT Scelta
      IF (Scelta < 1) OR (Scelta > 4) THEN PRINT : PRINT "Scelta errata"
    LOOP UNTIL (Scelta >= 1) AND (Scelta <= 4)
    LOCATE 12, 1
    IF (Scelta = 1) THEN PRINT "La somma vale : "; V1 + V2: PRINT
    IF (Scelta = 2) THEN PRINT "La sottrazione vale : "; V1 - V2: PRINT
    IF (Scelta = 3) THEN PRINT "Il prodotto vale : "; V1 * V2: PRINT
    IF (Scelta = 4) AND (V2 <> 0) THEN PRINT "Il quoziente vale : "; V1 / V2: PRINT
    IF (Scelta = 4) AND (V2 = 0) THEN PRINT "Divisione per zero!!": PRINT
    PRINT "Premi un tasto ('q' per finire) "
    DO
      Ch = INKEY$
    LOOP UNTIL Ch <> ""
  LOOP UNTIL Ch = "q"
  PRINT : PRINT "fine"
END
```

Il programma "tradotto" nel linguaggio QuickBasic

presenza e dovremo premere un tasto per esaminarne tipo e natura grazie ad una maschera che apparirà in basso sullo schermo.

Con i tasti cursore potremo quindi spostarci mentre, contemporaneamente, Turbo c visualizzerà, sulla restante area del video, la parte del programma sorgente nella quale è stato riscontrato l'errore.

Tramite il tasto **F1** potremo ottenere una completa descrizione dell'errore occorso. Tramite **F6** possiamo spostarci dalla finestra degli errori a quella di Edit, e viceversa. Tale possibilità permette di correggere gli errori commessi e, con immediatezza, posizionarsi nuovamente nella finestra degli errori per saltare velocemente da un errore ad un altro, correggendoli tutti prima di tornare al punto 6.

9) Per far girare il programma non dovremo far altro che premere i tasti **Alt + R**, selezionare **Run** e premere Enter, oppure sostituire l'intera sequenza con la pressione dei tasti **Ctrl** e **F9**. Turbo C "si accorge" che il programma è già stato compilato in precedenza, e quindi lo fa girare senza ricompilarlo. Effettuando una qualsiasi modifica, ovviamente, Turbo C compilerà nuovamente il programma prima di eseguirlo.

```
10 REM
20 REM OPERAZ.BAS
30 REM Esegue la somma, la sottrazione, il prodotto ed il quoziente tra 2 numeri
40 REM By Mariani Giancarlo 1990
50 REM Linguaggio : GWBASIC IBM o Commodore 64
60 REM
65  CH$=""
70  WHILE CH$<>"q"
80    CLS
90    PRINT "Introduci due numeri separati da una virgola e premi CR"
100   INPUT V1, V2
110   LOCATE 5, 1
120   PRINT "1 - Somma"
130   PRINT "2 - Sottrazione"
140   PRINT "3 - Prodotto"
150   PRINT "4 - Quoziente": PRINT
160   PRINT "Scegli : "
165   SCELTA=0
170   WHILE (SCELTA<1) OR (SCELTA>4)
180     LOCATE 10, 10
190     INPUT SCELTA
200     IF (SCELTA < 1) OR (SCELTA > 4) THEN PRINT : PRINT "Scelta errata"
210   :WEND
220   LOCATE 12, 1
230   IF (SCELTA = 1) THEN PRINT "La somma vale : "; V1 + V2: PRINT
240   IF (SCELTA = 2) THEN PRINT "La sottrazione vale : "; V1 - V2: PRINT
250   IF (SCELTA = 3) THEN PRINT "Il prodotto vale : "; V1 * V2: PRINT
260   IF (SCELTA = 4) AND (V2 <> 0) THEN PRINT "Il quoziente vale : "; V1 / V2: PRINT
270   IF (SCELTA = 4) AND (V2 = 0) THEN PRINT "Divisione per zero!!": PRINT
280   PRINT "Premi un tasto ('q' per finire) "
300   CH$=INKEY$:IF CH$="" THEN GOTO 300
320 :WEND
330 PRINT : PRINT "fine"
340 END
```

Il programma scritto nel "tradizionale" Gw-Basic

10) Turbo C, compilando un programma sorgente (dotato di suffisso **.C**), produce direttamente su disco la versione eseguibile (dotata di siffisso **.EXE**); supponendo, quindi, che il programma funzioni correttamente, non avremo più bisogno di restare in "ambiente" Turbo C per farlo girare.

Per uscire da Turbo C bisognerà premere i tasti **Alt + F**, selezionare l'opzione **Quit** e premere Enter, oppure premere direttamente **Alt + X**, che sostituisce la sequenza.

11) Impartendo, da Ms-Dos, il comando **Dir** noteremo che, oltre al sorgente **Operaz.C**, è presente il file oggetto **Operaz.OBJ** ed il file eseguibile **Operaz.EXE**. Per richiamare il programma sarà quindi sufficiente digitare, da Ms-Dos, **Operaz** e premere Enter.

Il programma eseguibile ottenuto non necessita, per un suo corretto funzionamento, della presenza di file del Turbo C e può essere liberamente copiato o trasportato su qualsiasi altro computer Ms-Dos, anche se non dotato di Turbo C.

Conclusioni

Siamo così giunti alla fine della nostra prima passeggiata nei giardini del C. Invitiamo i nostri lettori, come al solito, ad apportare semplici modifiche al programma pubblicato in modo da impadronirsi delle prime, fondamentali, regole sintattiche del potente compilatore. Abbiamo provveduto ad accludere anche una versione scritta in **QuickBasic 4.5**, allo scopo di far notare le differenze di sintassi tra l'obsoleto **Gw-Basic** ed un moderno e potente "interprete / compilatore" Basic. Si può facilmente notare come quest'ultimo si avvicini moltissimo ai linguaggi tradizionalmente strutturati, quali Pascal e, appunto, C, senza però raggiungerne la notevole velocità di esecuzione tipica dei veri compilatori.

MINI- GRAFICA PER IL C/128

La notevole potenza delle istruzioni grafiche consente di realizzare brevi, ma interessanti programmi

di Fulvio Collodi

Nel lontano n. 41 di C.C.C. fu pubblicato un programma grafico, per C/64, che richiedeva le (allora) famose routine grafiche di **Toma**.

In quella versione, l'autore R.

Morassi usò un algoritmo matematico, chiamato **Hopalong**, che realizzava interessanti figure tramite il continuo plottaggio di numerosi punti. Quella che pubblichiamo stavolta è la versione per **C/16, Plus 4(!) e C/128** del programma citato e presenta alcune differenze, sia in meglio che in peggio.

Anzitutto, al fine di ottenere la completa compatibilità dei computer menzionati, è stata tralasciata la possibilità di salvare e ricaricare le schermate; tuttavia adesso è possibile riferirsi ad un maggior numero di funzioni matematiche

```
100 rem * hopalong v2.0      c16/+4 & c128
110 rem * by collodi fulvio altopascio (lu)
120 :
130 color0,2:color1,1:color4,2
140 if(0)>0thenc=16:else c=128
150 forf=1to7:readf$(f):keyf,"":next:scnclr
160 f=1:s%=1:p%=1:printchr$(8)chr$(14)
170 char,11,2,"█ HOPALONG V2.0  █"
180 char,13,4,"C16/+4 & C128"
190 char,11,5,"by Fulvio Collodi"
200 char,11,6,"████████████████████"
210 char,11,11,"Def fn(x)? "+f$(f)
220 xx$="Parametri?"+str$(a%)+","
230 char,11,13,xx$+str$(b%)+","+str$(c%)
240 char,11,15,"Delta X-Y?"+str$(x%)+","+str$(y%)
250 char,11,17,"Scala X/Y?"+str$(s%)
260 char,11,19,"Dim punto?"+str$(d%)
270 char,11,21,"Loop step?"+str$(p%)
280 char,20,11,"":inputk$
290 forf=1to7:iff$(f)<>k$thennext:goto280
300 onfgosub630,640,650,660,670,680,690
310 char,20,13,"":inputa%,b%,c%
320 ifabs(a%)>99orabs(b%)>99orabs(c%)>99then310
330 char,20,15,"":inputx%,y%
340 ifabs(x%)>99orabs(y%)>99then330
350 char,20,17,"":inputs%
360 ifs%<1ors%>9then350
370 char,20,19,"":inputd%
```

L'Amiga può fare molto di più, ma il C/128, nel suo piccolo...

e, caratteristica assai importante, aumentare di non poco la velocità di plottaggio. All'inizio appare un menù relativo ad alcune informazioni:

1 - Tipo di funzione matematica da applicare, da indicare attraverso le relative iniziali (Sin, Cos, Tan, Atn, Log, Sqr, Exp).

2 - Gli incrementi ("Delta"), ovvero le traslazioni in pixel necessarie per centrare la figura.

3- La scala, ovvero la grandezza desiderata della figura.

4- La dimensione dei punti da plottare, sapendo che Dim = Input + 1.

5- Lo step, ovvero il "passo" da tenere drante la tracciatura del disegno.

Il plottaggio ha termine quando la variabile contatore **I**, incrementandosi dello step **S%**, supera **9999**, oppure quando l'utente preme un tasto.

A questo punto premendo il tasto **Esc** il programma finisce definitivamente, altrimenti si torna al menù iniziale.

Se durante il plottaggio viene premuto il tasto **F** si accede al modo **Fast** (anche sul C/16); così facendo la figura evolve più velocemente, scomparendo però alla vista.

Per farla ritornare occorre premere il tasto **S**: la figura apparirà nuovamente, ma la velocità di plottaggio, da quel momento, continuerà a velocità ridotta (modo **Slow**).

Il diverso trattamento dei modi Fast e Slow del C16/+4 dal C/128, obbliga il programma a riconoscere la macchina su cui gira.

Tale riconoscimento avviene all'inizio, tramite l'uso della funzione **Rlum**, non disponibile sul C/128 e quindi considerata come un vettore il cui contenuto sarà sempre nullo.

Grafincos

Anche questo breve programma genera figure in alta risoluzione applicando a tre parametri le funzioni matematiche **seno** e **coseno**.

In ogni figura è possibile riscontrare un cerchio centrale; il parametro **RC** esprime il raggio di tale cerchio.

I restanti parametri, **R1** e **R2**, stanno ad indicare la lunghezza delle linee tracciate; la loro

I listati pubblicati possono essere migliorati dal lettore

```
380 ifd%<0ord%>2then370
390 char,20,21,"":inputp%
400 ifp%<1orp%>100then400
410 i=0:x=0:y=0:dx=x%+159:dy=y%+99
420 scale0:graphic1,1:sshapep$,0,0,d%,d%
430 char,i,i,str$(a%)+str$(b%)+str$(c%)
440 char,11,i,"y="+f$(f)+"(x)"
450 char,20,i,str$(x%)+str$(y%)
460 char,27,i,str$(s%):char,31,i,"loop"
470 do:getk$:i=i+p%:char,35,0,str$(i)
480 gshape p$,(y+x)*s%+dx,(y-x)*s%+dy,1
490 x1=y-sgn(x)*fny(x):y1=a%-x:x=x1:y=y1
500 loopuntili+p%>9999ork$<>""
510 ifk$="f"ork$="s"thengosub570:goto470
520 gosub600:char,31,0,"esc = end",1
530 do:getk$:loopwhilek$="":graphic0,1
540 ifk$=chr$(27)thenend:else170
550 :
560 datacos,sin,tan,atn,sqr,log,exp
570 ifk$="s"then600
580 ifc=16thenpoke65286,11:elsefast
590 return
600 if c=16 then poke65286,27:else slow
610 return
620 :
630 deffny(x)=cos(abs(b%*x-c%)):return
640 deffny(x)=sin(abs(b%*x-c%)):return
650 deffny(x)=tan(abs(b%*x-c%)):return
```

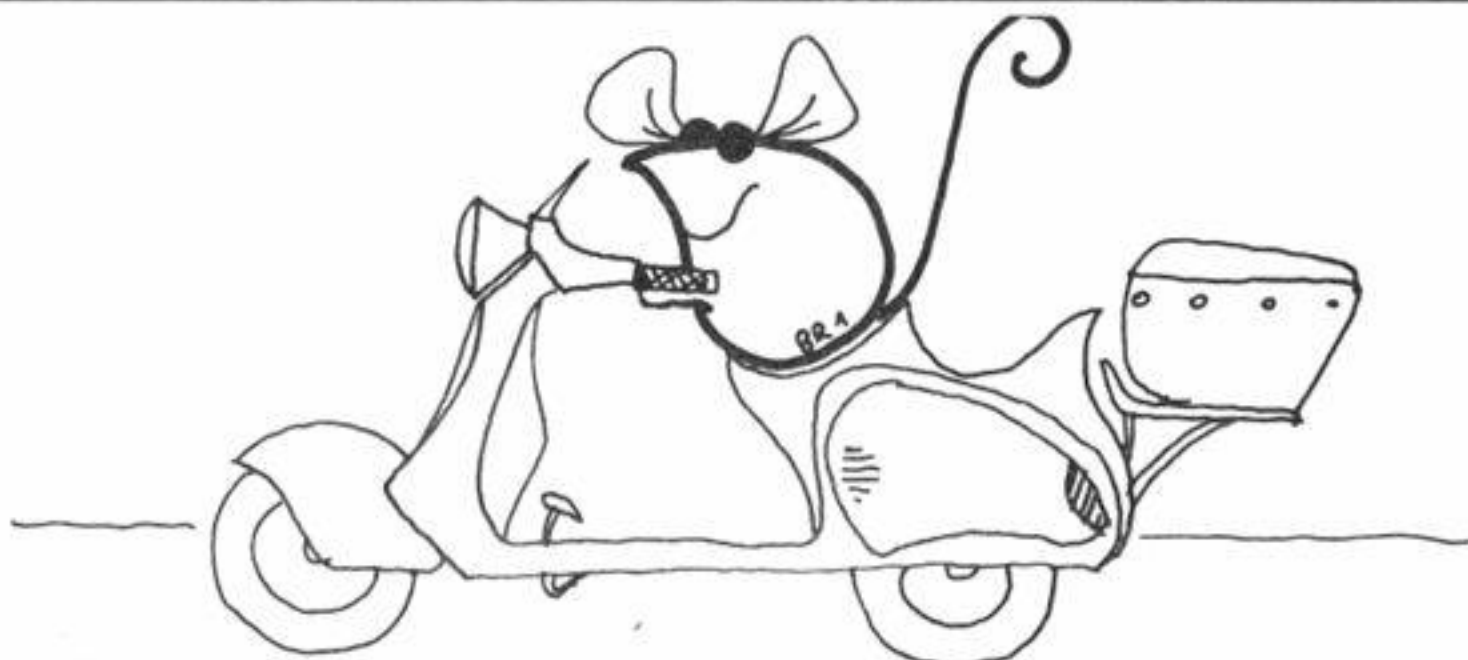
```
660 deffny(x)=atn(abs(b%*x-c%)):return
670 deffny(x)=sqr(abs(b%*x-c%)):return
680 deffny(x)=log(abs(b%*x-c%)):return
690 deffny(x)=exp(cos(b%-x-c%)):return
700 end
ready.

100 rem * grafin-cos  c16/+4 & c128
110 rem * collodi fulvio  altopascio (lu)
120 graphic0,1:scale0:color1,1:color0,2
130 input" rc,r1,r2";r%,r1%,r2%:x=0
140 gosub 280
150 char,0,0,str$(r%)+str$(r1%)+str$(r2%)
160 do:x=x+.15:getk$:ifk$<>""thenexit
170 x1=160+cos(x)*r%:y1=100+sin(x)*r%
180 x2=x1+(r1%+r2%)*sin(x*r2%/r1%)
190 y2=y1+(r1%+r2%)*cos(x*r2%/r1%)
200 ifx<.2thenp1=x2:p2=y2:goto230
210 draw,zx,zytox1,y1tox2,y2toxx,yy
220 ifabs(p1-x2)<[illegible]andabs(p2-y2)<[illegible]thenexit
230 xx=x2:yy=y2:zx=x1:zy=y1
240 loop:char,30,0,"esc = end"
250 do:getk$:loopwhilek$=""
260 ifk$=chr$(27)thengraphic0,1:else120
270 end
280 xx=0
290 if abs(r1%)<100 then xx=xx+1
300 if abs(r2%)<100 then xx=xx+1
310 if abs(r%)<100 then xx=xx+1
320 if r1%<>0 and xx>2 then graphic1,1:else120
330 return
340 end
```

Alcuni paramtri possono generare errori; procedete per tentativi

proporzionalità comporta gli effetti più interessanti. Il plottaggio termina premendo un tasto, od automaticamente quando la figura ha raggiunto una certa simmetria.

A questo punto, premendo **Esc**, il programma finisce definitivamente, altrimenti si torna nuovamente all'immisione di nuovi parametri, che devono comunque assumere valori interi compresi tra **-100** e **+100**, escluso lo zero per il parametro **R1**.

Il computer si accorge di eventuali immissioni errate e provvede ad una nuova richiesta.

CAMPUS

AMIGA

SOMMARIO

66 - SYS 49152, ANCHE CON AMIGA

Chi passa dal C/64 all'Amiga prova un grandissimo imbarazzo quando decide di imparare l'Assebly. Sul vecchio computer ad 8 bit, infatti, era possibile allocare i codici macchina in una zona libera della memoria, di solito posizionata a partire dalla storica locazione 49152.
Con Amiga, purtroppo (o per fortuna?) non è dato di sapere con precisione una zona libera di Ram in cui allocare con sicurezza i codici in linguaggio macchina. Sia per la struttura stessa di Amiga, sia per il multitasking, il "sistema" colloca un'eventuale routine dove gli pare ed è difficilissimo individuarla allo scopo di attivarla. Grazie ad un trucchetto, spiegato nei dettagli, è tuttavia possibile, da Basic, non solo allocare una routine, ma addirittura(!) lanciarla.

69 - AMIGAFACILE

Continua la carrellata di spiegazioni sul corretto uso dei comandi Dos. Vengono esaminati altri files vitali del Sistema Operativo di Amiga e proposti nelle varie sintassi possibili.
Ove necessario, alcuni files batch contribuiscono a fare chiarezza sulle avare notizie reperibili nello scarno manuale del computer.
Diverse forme sintattiche insolite, e l'elenco degli errori più frequenti, eviteranno di far perdere tempo a chi desidera approfondire i "misteri" di AmigaDos.

76 - POSTAMIGA

Questo mese la tradizionale rubrica dedicata ai dubbi dei nostri affezionati lettori diventa parte integrante di Campus, dal momento che gran parte delle domande riguardano il corretto uso di comandi del Dos o il funzionamento dei drives (che è quasi la stessa cosa).
Un po' per volta PostAmiga tende a diventare, grazie soprattutto alla competenza di Domenico Pavone, una vera e propria enciclopedia a dispense.

Nessun albero viene abbattuto per gli inserti di Commodore Computer Club, stampati su carta riciclata al 100%

SYS 49152: ANCHE SU AMIGA

Il trasferimento di parametri dal Basic all'Assembler, il famoso "Sys 49152, X" del C/64, è possibile anche nel meraviglioso Amiga

di Gianluigi Mongillo

Può far sempre comodo inserire una routine l.m. in propri programmi Basic

L'Amiga, come è noto, è un computer **multitasking**, capace cioè di eseguire più programmi (=Task) contemporaneamente; per questa ragione non è disponibile un'area di memoria Ram, come la zona del C/64 da $C000 a $CFFF, in cui depositare stabilmente dati e programmi.

D'altronde, se ci fosse un posto simile, gli utenti desiderebbero memorizzarvi le proprie routine, con la conseguente conflittualità o perdite di dati che ogni 64-ista ben conosce.

I programmi in linguaggio macchina (l.m.) è bene che siano **Position - Indipendent**, devono cioè essere caricati a partire da qualsiasi indirizzo oppure, meglio, da un indirizzo che venga volta per volta automaticamente calcolato dallo stesso elaboratore.
Non è molto difficile rendere i propri programmi l.m. **rilocabili**; basta, infatti, utilizzare l'indirizzamento relativo al **P.C.** (=**P**rogram **C**ounter, contatore di programma), uno dei 13 tipi di indirizzamento che il cervello siliceo di Amiga mette a disposizione. In parole povere, in un programma rilocabile è indispensabile evitare istruzioni del tipo...

MOVE.L #BUFFER, A1

...oppure...

CMPI.B #13, INBUF

...che vanno invece diligentemente trasformate in:

LEA.L BUFFER(PC), A1

...e...

LEA.L INBUF(PC), A1
CMPI.B #13, (A1)

L'indirizzamento relativo al P.C. permette solo una distanza di **16** bit preceduta da un segno (+/- 32 KByte).

Per eseguire una routine assembler occorre che essa venga (lapalissiano!) caricata in memoria e che i suoi codici siano salvati in una stringa, o in una matrice, opportunamente dimensionata.

La matrice deve essere necessariamente di tipo **Integer**, altrimenti l'interprete Basic trasformerebbe i valori interi in numeri a virgola mobile: del nostro codice rimarrebbe un lontano ricordo.

Ancora, nel caso si utilizzi la matrice, i dati devono essere immagazzinati necessariamente sotto forma di **Word**, poichè, se registrati come semplici **Byte**, la parte alta del numero sarebbe azzerata; anche in questo caso addio codice!

Il programma Basic n.1 (che pubblichiamo solo a titolo di esempio) ha le seguenti funzioni: il file in linguaggio macchina ha il nome di **Prog** e si trova nella subdirectory **Laura** del drive **Df0:**. Il programma Basic ne legge i dati e li memorizza nella stringa **Ass$**, poi ricava l'indirizzo di partenza e richiama la routine.

```
CLS
'Attenzione: questo programma è solo dimostrativo, non funzionante!'
OPEN "DF0:LAURA/PROG" FOR INPUT AS #1
L=LOF(1)
ASS$=INPUT$(L,1)
CLOSE #1
ASS&=SADD(ASS$)
CALL ASS&
END
```

Programma n.1

Questo metodo è semplice, veloce e molto utile in fase di Debbuging, visto che il file L.M. può essere caricato con un qualsiasi Monitor / Debugger e modificato a piacimento. A patto, però, che sul di-

sco siano presenti entrambi i moduli (Basic e LM). La soluzione più semplice è quella di incorporare il codice sotto forma di **Data** all'interno del programma Basic, come più volte accennato sulle pagine di C.C.C. Ma come si fa a realizzarlo in pratica?

Risposta: mediante uno dei tantissimi programmi di pubblico dominio atti a questo scopo, e se per caso non l'avete, non resta che digitare il programma **Store** riportato in queste pagine.

Ci si può render conto che ogni volta che viene richiamata, la routine viene caricata in una posizione diversa, ed ecco perchè è buona norma che le routine destinate ad interagire con il Basic siano Position - Indipendent.

Ma ora veniamo all'argomento principale dell'articolo e cioè lo scambio di parametri tra Basic e Assembler e viceversa. Il comando Basic **Call** ha due forme sintattiche:

CALL CodeAdr&

...e...

CALL CodeAdr& (Lista - Parametri)

L'affare si complica

In che modo i parametri della lista vengono passati alla routine LM? Sul manuale di AmigaBasic, a pagina115, si legge:

I parametri vengono passati tramite un valore utilizzando le convenzioni del Linguaggio C

Non sembra abbastanza chiaro, quindi cerchiamo di approfondire con semplici esempi.

E' bene sapere che i parametri delle funzioni del **C** vengono passati allo **Stack** e poi, con grande spreco di codici, dallo Stack ai registri utilizzati dalla funzione. Da ciò avrete certamente capito che i parametri vanno a finire nello Stack.

Lo stack è una struttura LIFO (=**L**ast **I**n **F**irst **O**ut e cioè l'ultimo dato ad entrare è il primo ad uscire); per capire meglio aiutiamoci con una analogia:

Si immagini una pila di piatti, disposti su un "distributore" di vassoi a scatto, comuni nelle mense, fatti in modo che il piatto in cima resti sempre al livello del bancone.

Appena si prende un piatto, quelli posti sotto vengono

```
CLS
PRINT "Object Code Basic Data"
PRINT "by Gianluigi Mongillo"
PRINT "1990"
PRINT "Vedi C.C.C. #74 Pag. 74"
start:
PRINT
INPUT "Source File: ",sf$
INPUT " Data File: ",df$
CLS
PRINT "File Oggetto";sf$:
PRINT "File di DATA in Basic";df$
PRINT:PRINT "OK? (S/N) "
loop:
a$=INKEY$:IF a$="" THEN loop
a$=UCASE$(a$)
IF a$="N" THEN start
BEEP
OPEN sf$ FOR INPUT AS 1
dati$=INPUT$(LOF(1),1)
CLOSE 1
OPEN df$ FOR OUTPUT AS 1
totdati=LEN(dati$)
PRINT #1,"' Dati = ";totdati
dpl=10
numlinee=INT(totdati/dpl):ultimo=dpl-1
FOR linee=1 TO numlinee*dpl STEP dpl:PRINT #1,"DATA ";
 FOR dati=0 TO ultimo
  a=ASC(MID$(dati$,linee+dati,1)):GOSUB store
 NEXT dati
NEXT linee
IF numlinee >< totdati/dpl THEN
 ultimo=totdati-numlinee*dpl:PRINT #1,"DATA ";
 FOR dati=1 TO ultimo
  a=ASC(MID$(dati$,(totdati-ultimo)+dati,1)):GOSUB store
 NEXT dati
END IF
PRINT:PRINT "Fine":END
store:
a$=STR$(a)
IF a<10 THEN PRINT #1,CHR$(48);
IF a<100 THEN PRINT #1,CHR$(48);
PRINT #1,RIGHT$(a$,LEN(a$)-1);
IF dati=ultimo THEN PRINT #1,"" ELSE PRINT #1,",";
```

Il programma "Store"

La procedura descritta si può facilmente rendere automatica

La funzione "Merge" del Basic permette un facile trasporto di routines in l.m.

```
CLS
ASS$ = ""
'CARICA IL CODICE
FOR I=0 TO 27
 READ DATO
 ASS$=ASS$+CHR$(DATO)
NEXT I
ASS&=SADD(ASS$)
A$="HELLO WORLD!"
'PREPARA PARAMETRI PER LA ROUTINE
INDIRIZZO&=SADD(A$)
LUNGHEZZA%=LEN(A$)
CALL ASS&(INDIRIZZO&, LUNGHEZZA%)
'RICHIAMA LA ROUTINE
PRINT A$
'MOSTRA LA VARIABILE MODIFICATA
'DATI DEL CODICE
' Dati = 28
DATA 072, 231, 128, 128, 032, 111, 000, 012, 032, 047
DATA 000, 016, 083, 128, 017, 188, 000, 120, 008, 000
DATA 076, 223, 001, 001, 078, 117, 000, 000
```

Il programma n. 2

spinti in alto e, quando se ne aggiunge uno, tutti gli altri vengono spinti in basso.

I piatti si possono paragonare ai dati presenti nello Stack: aggiungendone uno, viene "sepolto" dagli altri dati posti successivamente. Il primo elemento può esser preso facilmente ma agli altri elementi si può accedere solo quando si è spostato ciò che si trova sopra di loro. Vediamolo con un esempio:

A% = 13
B% = 15
CALL CodeAdr&(A%,B%)

Sullo stack avremo:

(SP) Indirizzo di RETURN
4(SP) Valore di B% e cioè 15
8(SP) Valore di A% e cioè 13.

Ne sappiamo abbastanza; dopo questa grigia teoria passiamo alla pratica digitando, con il Seka Assembler, il programma **Esempio**.

E' solo un piccolo dimostrativo assembler che sostituisce con un carattere "**X**" l'ultimo carattere di una stringa; per fare ciò dovremo dare due parametri, cioè l'**indirizzo** della stringa e la sua **lunghezza**.

Assembliamo il programma; poi, al riapparire del prompt Seka, battendo il comando **O** sul video verrà stampata una tabella di cui interessa solo l'ultima voce (**Code**) che mostra l'indirizzo iniziale del codice, quello finale e il numero di bytes occupati dal programma stesso.

Passiamo ora a salvare il codice grezzo, utilizzato dal programma Store, su disco: battere il comando **WI** e rispondere con i dati osservati alla voce Code alle domande Begin ed End.

A questo punto avremo sul disco il file LM (non eseguibile direttamente da CLI poichè manca l'OverHead) a cui avremo assegnato, per esempio, il nome **Laura**.

```
MOVEM.L A0/D0, -(SP) ;8 in più sullo stack + return address = 12
MOVE.L 12(SP), A0 ;preleva l'indirizzo della stringa (4 byte)
MOVE.L  16(SP), D0 ;preleva la lunghezza della stringa (4 byte)
MOVE.B #'X', -1(A0, D0) ;sotituisce l'ultimo carattere con una X
MOVEM.L (SP)+, A0/D0 ;ripristina il valore dei registri A0 e D0
RTS ;ritorna al basic
```

Il programma "Esempio"; sintassi Seka

Entriamo in ambiente Basic e lanciamo Store; alla prima domanda risponderemo con il nome prima assegnato (=Laura), alla seconda con un nome di nostro gradimento che indichi il file che conterrà il codice sotto forma di Data; in tutti i casi aggiungeremo eventualmente il percorso da seguire nel caso si decida di usare subdirectory.

Non resta che unire (con Merge) il modulo Data al programma principale e il gioco è fatto (vedi programma 2).

POSTAMIGA

(a cura di Domenico Pavone)

Rename diventa move

Mi sono imbattuto in un problema un po' strano con il comando Rename del Dos. Volevo trasferire una serie di files del mio dischetto di lavoro in un'altra directory, dove era presente una vecchia copia degli stessi files che non mi interessava più. Rename, però, si rifiuta di funzionare se un file da spostare esiste già nella nuova sede, e ho dovuto così adoperare Copy e poi Delete per cancellare gli originali. C'è un modo per evitare queste manovre adoperando il solo Rename?

(Claudio Rinaldi - Salerno)

Nella rubrica Amigafacile (n. **76** della rivista), a proposito del comando Rename, viene proprio riportato questo tipo di problema, che può insorgere quando si tenta di adoperare Rename per spostare un file da una directory all'altra.

In effetti non è un caso troppo frequente, e comunque non va sottovalutata la prudenza dimostrata dai progettisti del Dos nell'evitare indesiderate sovrascritture di files, con relativa perdita del precedente contenuto.

Come si diceva in quella sede, tuttavia, risulterebbe sicuramente più comodo qualcosa di simile ad un requester che richieda una conferma sull'eventuale cancellazione del file preesistente; per ottenerlo, sarebbe però necessario riprogrammare il comando.

Qualcosa tuttavia si può fare, e con risultati niente affatto disprezzabili, sfruttando al massimo quanto viene offerto dal Dos di Amiga.

Per esempio si potrebbe creare un nuovo comando, **Move**, che si occupi solo degli spostamenti di files, con la possibilità di sovrascrivere (previo conferma) eventuali copie preesistenti. La cosa è più semplice di quanto si creda: tutto ciò che occorre, è un (quasi) banale **Batch File**.

Per rendere il tutto di pratica ed immediata esecuzione, si proceda come segue.

Ci si porti nella directory **C** del disco di sistema (anche il Workbench va bene, purchè abbia qualche decina di byte liberi) impartendo, da Shell, il comando **Cd C**.

Adoperando l'editor **Ed**, si crei un file batch a nome Move (comando **Ed Move**) così strutturato:

```
.key origin,destin
if exists <destin>
echo "Il file esiste già!"
ask "Sovrascrivo? (y/n)"
if not warn
quit
endif
endif
delete <destin>
rename <origin> <destin>
```

Si esca ora da Ed con la sequenza **Escape**, tasto **X** e Return, e si impartisca da Shell un comando **Protect Move + S**. Si badi che, nel caso si dovesse ri-editare il file (magari per correggerne un errore), si renderebbe poi indispensabile reimpartire il comando Protect appena visto, in quanto Ed resetta lo stato dei bit di protezione associati al file.

Tutto fatto. Da questo momento si disporrà di un nuovo comando, che poi non è veramente tale, ma che funzionerà come qualunque altro, adoperando la sintassi...

Move Nome1 Nome2

Il parametro **Nome1** deve indicare nome e percorso completo del file da spostare, mentre **Nome2** specifica il percorso (sempre comprensivo del nome) di destinazione. Se nella directory di destinazione risulterà già presente un file di quel nome, verrà visualizzata una richiesta di conferma sull'opportunità o meno di sovrascrivere il file; in caso contrario, tutto procederà esattamente come per un normale Rename.

Attenzione al fatto che, non esistendo un controllo di errore, se nel parametro Nome2 si precisa un percorso (per esempio solo **Ram:**, senza alcun nome) non accettato da Rename, si attiva ugualmente la segnalazione di file esistente, ma, come ovvio, non si produrrà alcun effetto pratico.

La rubrica "PostAmiga" è quasi interamente dedicata al corretto uso dei comandi del Dos. Si consiglia la rilettura dei precedenti fascicoli di C.C.C.

Perchè Move funzioni correttamente, è necessario che nella directory C siano presenti i comandi **Execute** e **Rename**, nonchè **If, Endif, Quit, Ask** ed **Echo**. Lo si tenga presente nel caso lo si trasferisca altrove: anche se ne simula l'efficacia, Move non è un programma autosufficiente, ma un semplice batch file!

Con le manovre prima affrontate, ne abbiamo però settato il bit di protezione Script, per cui può essere mandato in esecuzione semplicemente invocandone il nome. Considerata questa supercomodità, il file è stato inoltre memorizzato nella directory **C**, piuttosto che la tradizionale **S**, in modo da essere incluso nel normale path dei comandi Dos, e quindi "raggiungibile" **quale che sia la directory corrente**.

Quanto al funzionamento del batch, questo può essere così riassunto: con **.KEY** vengono prelevati i parametri che seguiranno il comando. Ogni qualvolta, all'interno del file, vengono adoperati **<origin>** e **<destin>**, questi saranno sostituiti da quanto digitato nella sintassi di Move.

Il comando **Ask** consente l'input di un carattere (**Y** = yes, **N** = no), e relativa redirezione del flusso operativo in base alla condizione **Warn**.

Facile il resto: se era già presente un file dello stesso nome (**exists**), e si è data risposta affermativa all'input, questo viene cancellato prima di eseguire il vero comando rename.

Per velocizzare il tutto, si può evitare l'uso di Echo, magari inserendone il testo (in un'unica riga) nella stringa associata al comando Ask.

Questo Move, essendo pur sempre legato a Rename, consente lo spostamento di un file solo nell'ambito dello stesso dischetto. Se lo si volesse rendere più versatile, basterà sostituire le ultime due righe del batch in modo che diventino...

copy <origin> <destin>
delete <origin>

In questo caso, si potrà trasferire il file anche in altri dischi o Ram Disk, mentre nella directory C si renderà obbligatoria la presenza del comando Copy oltre a quelli in precedenza elencati.

Ultima avvertenza: viene dato per scontato che si adoperino i comandi (soprattutto protect) nella loro **Versione 1.3**, e che si adoperi la Shell, **non** Cli.

In quest'ultimo caso, infatti, il comando Move, se impartito direttamente, non verrebbe riconosciuto come script, e si renderebbe necessario ricorrere ad un più "verboso" **Execute C/Move Nome1 Nome2**.

Al di là del pur comodo Move, con la tecnica affrontata è possibile implementare un numero infinito di varianti ai comandi del Dos.

Con un pizzico di fantasia, ma accompagnata da un minimo di approfondimento.

Amigabasic alle corde

Chiedendo la directory del disco da basic, con Files "dfx:", i nomi dei programmi e quelli con suffisso ". Info" vengono visualizzati senza criterio. Esiste un modo per separare le due categorie per rendere più semplice la ricerca?
(Gianluca Rizzi - Golasecca)

Risposta inequivocabile: no. O almeno, non direttamente dalla finestra di output del basic.

Unica alternativa possibile, è quella di mantenere aperta in multitasking anche una finestra Shell o Cli, quindi agire sui gadget di profondità delle due finestre di Amigabasic per portarle in secondo piano.

A questo punto, si potrà impartire da dos il comando...

List #?.info

...per visualizzare solo i files-icona (il comando Files del basic non supporta le wild cards, o caratteri speciali che dir si voglia), oppure **List Files** per i programmi veri e propri, o ancora **List Dirs** per selezionare i soli nomi di directory.

Ottenuta l'informazione, si potrà poi tornare al basic portando "indietro" la finestra Shell.

Il tutto, naturalmente, annegando nelle esasperanti attese dei "refresh" di schermo del basic.

Anzi, di AmigaBasic, ormai senza ombra di dubbi da considerare come il peggior interprete tra quelli in circolazione, se non verrà aggiornato da qualche nuova release (eventualità che sembra remota).

Unico punto in suo favore, quello di essere fornito in dotazione al computer, mentre altri e ben più validi concorrenti costringono a qualche esborso in più.

Tornando in argomento, dunque, non resta che arrangiarsi: è già tanto che un comando Files esista!...

File al microscopio

Vorrei porvi tre quesiti, cui spero vogliate rispondere senza rinviarmi al vostro servizio arretrati:
1) Impartendo da Shell il comando Type Nomefile H, ho notato che il file viene mostrato disposto su sei colonne, in notazione esadecimale. A che cosa corrisponde ognuna di quelle sei colonne, e come si possono editare?
2) C'è un modo, in basic, per leggere il solo nome della directory del disco presente in un drive?
3) Come posso mandare in esecuzione un programma del C/64 trasferito con l'Emulator su un disco Amiga, se questo non ha nemmeno l'icona?
(Paolo Giovannelli - Lavinio)

Intanto, cominciamo col dire che la forma sintattica **Type Nomefile H** non produce affatto il "dump" esadecimale del contenuto di un file, bensì una sua riscrittura in un file di

nome "**H**" che si troverà, poi, nella directory corrente.

La forma corretta è quindi...
Type Nomefile opt h

Per lo più, il comando Type viene adoperato per visualizzare dei file Ascii. Adottando la sintassi appena descritta, è invece possibile adoperarlo su qualunque tipo di file, compresi i cosiddetti eseguibili.

Nell'output che ne risulta, la prima "colonna" a sinistra (in realtà è il raggruppamento di 4 colonne video) indica il cosiddetto **offset**, ovvero la posizione relativa all'interno del file stesso di ogni suo byte, in notazione esadecimale.

Più chiaramente: se (per esempio) la prima colonna riporta **0060:**, significa che il primo byte che segue, nella stessa riga, (continuando ad adottare la terminologia un po' impropria del lettore, diremmo nella seconda "colonna") è il 97-simo byte del file. Quello subito a destra sarà il 98-esimo, e così via.

Si badi, però, che il valore esadecimale **0060** corrisponde, in decimale, a **96**. Il fatto che il byte sia il 97esimo e non il 96esimo è legato al conteggio dell'offset, che inizia da **zero** e non da **uno** (Primo byte = byte zero).

I quattro raggruppamenti di valori che seguono l'offset, rappresentano il codice numerico del file, rappresentato in **Long Word**, ovvero 4 byte alla volta (per un totale di **16**). Per chi è ancora alle prime curiosità, va detto che ogni byte è rappresentato da due cifre, sempre in esadecimale, che, per capirci meglio, nel caso di un file di testo corrisponderebbe al codice Ascii di un singolo carattere.

Proprio l'ultima "colonna" mostra la rappresentazione Ascii del file, con ogni riga che va riferita ai corrispondenti 16 valori esadecimali (due cifre per byte, si ricordi!) mostrati sulla stessa linea.

Quando si applica Type con l'opzione **H** ad un file non di testo, in questa sezione dell'output vengono raffigurati in Ascii solo i codici realmente visualizzabili, tutti gli altri (quelli il cui codice Ascii non corrisponde ad un simbolo grafico) verrano sostituiti da un punto.

Chiaro che, nei file programmi, si avrà quasi sempre un output privo di significato, in quanto i valori numerici rappresentano il "codice macchina" del file.

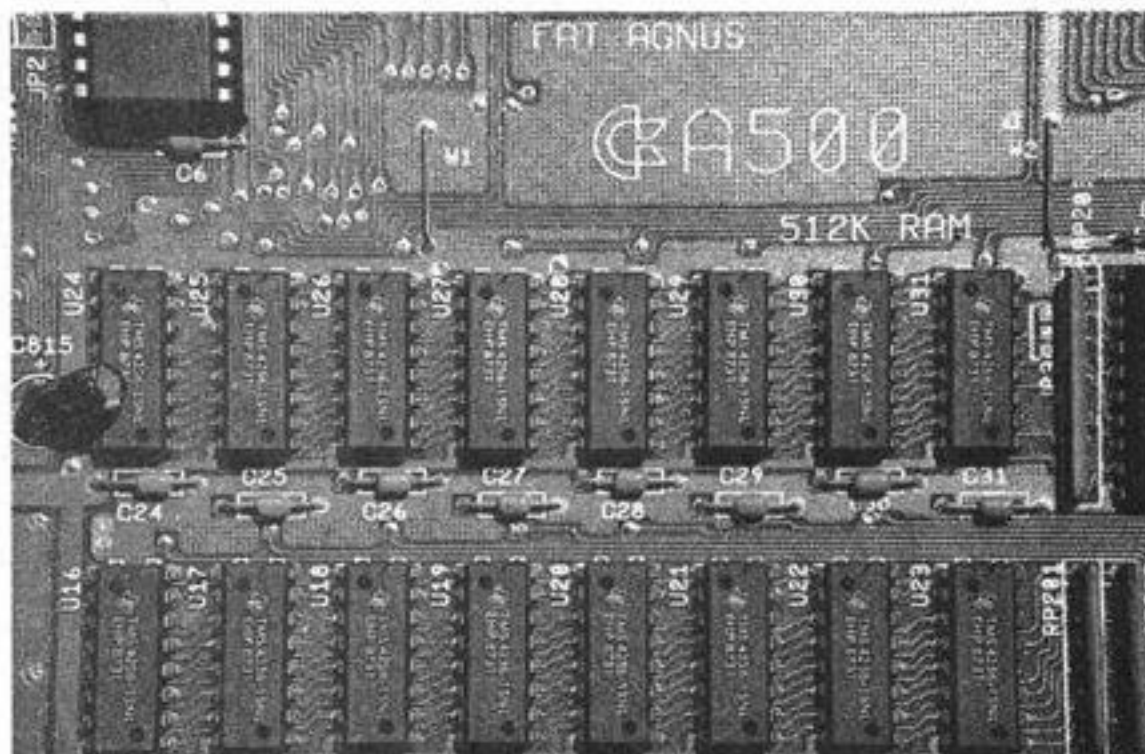
Ma qui il discorso si fa troppo ampio, e va rimandato ai vari articoli che si occuperanno di Linguaggio Macchina e cosucce simili.

L'editing diretto dei singoli byte di un file non è implementato da Type, e comunque è piuttosto sconsigliabile se, come traspare dalle domande del lettore, si è ancora alle prime armi: una modifica indiscriminata alla struttura di un programma, ne potrebbe decretare la definitiva... dipartita.

Chi volesse comunque cimentarsi, può sempre ricorrere ad alcuni programmi di pubblico dominio come **New-Zap** e **FileMaster**, mentre l'optimum, come ovvio, lo si raggiunge attraverso l'uso di più completi **Monitor / Debugger**.

Quanto alle altre domande, nonostante la buona volontà (...), è impossibile non rimandare il tutto a quanto già pubblicato proprio sul n. **77** della rivista (rubrica Postamiga).

Per leggere il nome della directory di un disco da basic, si veda, infatti, la procedura (ed il listato) descritta a proposito della differenza tra dischi apparentemente identici: escludendo i problemi di data, vi si può rintracciare tutto il necessario.

A proposito del (cattivo) rapporto tra Amiga e C/64, argomento tirato in ballo anche da **Annamaria Rampinelli** e **Amedeo Brunetti**, si veda il "decalogo" pubblicato dal numero scorso, che interesserà anche (per altri versi) il neo- futuro utente **Marco Alzetta** (lo spazio è tiranno!).

Il lettore si metta comunque l'anima in pace: un programma del C/64 **non** può girare autonomamente su Amiga, indipendentemente dalla presenza o meno di icone.

Provare, però, non nuoce. Quale migliore perversione del godere la vista di una splendida banda nera, abbellita da un riquadro rosso, con un poetico messaggio senza ritorno, di sapore orientale: Gur.....

Memoria e A-590

Penso di acquistare un hard disk A-590 per il mio Amiga 500. Vorrei sapere se i due Megabyte di Ram sono opzionali, e se servono solo per il disco rigido.
(Luca Mineo - Palermo)

Sì, l'espansione Ram dell'A-590 è opzionale. Sulla sua scheda di controllo sono presenti gli zoccoli sui quali *eventualmente* innestare gli integrati da 256 Kbit fino ad un ammontare di due megabyte.La ram aggiuntiva viene poi vista dal sistema come **memoria Fast**, utilizzabile quindi da qualunque applicazione che faccia uso di tale tipo di memoria, non solo in rapporto all'attività dell'hard disk.

A tutto ANSI

Ho letto che è possibile abilitare i vari modi testo (diverso colore, neretto, sottolineato, eccetera) da Cli adoperando delle sequenze di Escape. Riesco però ad usare le varie modalità solo impartendo la sequenza direttamente. Come posso fare per utilizzarle nei file creati con l'Ed?
(Alberto Verri - Somma Lombardo)

Per chi non lo sapesse ancora, ricordiamo che una "sequenza di escape" consente di immettere particolari codici, secondo uno standard chiamato **Ansi**, che possono (tra l'altro) modificare alcune caratteristiche grafiche di una finestra Dos, come (per esempio) il rapporto **colore** tra sfondo e primo piano e lo **stile** del testo.

Una sequenza di Escape è caratterizzata dall'uso dell'omonimo tasto (**Esc**), seguito da una parentesi quadra aperta (**[**), e dai codici Ansi che interessano.

Sequenza	*Sequenze di escape di uso piu' frequente* **Azione**
Esc[0m	Ripristino modo precedente
Esc[1m	Testo in neretto (Bold)
Esc[3m	Testo in corsivo (Italic)
Esc[4m	Testo sottolineato
Esc[7m	Testo in Reverse
Esc[30m	Colore testo = default
Esc[31m	Colore testo = bianco
Esc[32m	Colore testo = nero
Esc[33m	Colore testo = arancio
Esc[40m	Colore sfondo = default
Esc[41m	Colore sfondo = bianco
Esc[42m	Colore sfondo = nero
Esc[43m	Colore sfondo = arancio

A ridosso di questi ultimi, va poi aggiunto un carattere **"m"**, rigorosamente in **minuscolo** (si veda tabella pubblicata a parte).

Come già sperimentato dal nostro lettore, un uso diretto di queste sequenze non pone alcun problema. Si provi, per esempio, ad aprire una finestra Shell dal disco Workbench, quindi a pressare nell'ordine il tasto **Esc** (apparirà un paretesi quadra in reverse), il tasto **"["** (va bene anche quello del tastierino numerico), quindi **"4m"** e **Return**.

Il Dos segnalerà un **Unknown command**, ma dalla riga successiva il testo verrà visualizzato sottolineato. Per tornare alle condizioni precedenti, in questo caso quelle di default, basterà una nuova sequenza **Esc[0m**, o anche (non standard) un semplice Escape, il tutto seguito sempre dalla pressione del Return.

Si badi che, nel caso si utilizzi il Cli e non la Shell, il comportamento sarà leggermente diverso (lo si verifichi dopo aver impartito Newcli): identico il risultato, ma la sequenza di Escape non verrà visualizzata mentre la si digita, per cui sarà necessario procedere un po' alla cieca. Inoltre, il "trucco" del solo Escape per tornare alle condizioni iniziali non sortirà alcun effetto.

Queste differenze sono più importanti di quanto si creda, in quanto Shell consente alcune "performance" impossibili per il Cli (e addirittura anche per ED, come vedremo tra breve).

Tornando in argomento, c'è da aggiungere che in una stessa riga di comando è possibile associare una sequenza Escape multipla, ovvero che abiliti più caratteristiche. Ad esempio, si digiti (solo da Shell!) questa riga (per ESC si intenda la pressione dell'omonimo tasto):

ESC[33mESC[41mESC[1m

Dopo il canonico return, il testo successivo verrà visualizzato in grassetto, e di color arancio su sfondo bianco (per i codici, vedi tabella). Per tornare a condizioni normali, si ripeta ora la sequenza **ESC[0m**.

Fin qui si è visto come i codici Ansi possono essere utilizzati in modo diretto, ma ai fini di un uso più evoluto risulta decisiva la possibilità di inserirli nel contesto di un comando del Dos che preveda una qualche forma di output.

Bene, niente di più semplice, almeno in apparenza. Si provi a digitare, sempre all'interno di una finestra Shell, questa riga (ESC = tasto escape)...

Echo "ESC[33;41;1m Computer Club ESC[0m"

...senza omettere i doppi apici e soprattutto le parentesi quadre (occhio, possono confondersi con quelle in reverse che rappresentano l'Escape).

Stavolta si è ottenuto qualcosa di meno grossolano: dopo il Return, viene stampata a video la stringa, ma con le caratteristiche estetiche da noi assegnatele tramite la sequenza di escape. Si noti, tra l'altro, come sia possibile adoperare una sola volta l'Escape, facendolo poi seguire dai vari codici separati dal simbolo punto e virgola.

Per inciso, la stessa sequenza non funziona se impartita in modo diretto: occorrerebbe inserire un Escape per ogni singolo codice.

Da qui all'inserire le nostre sequenze (per esempio) in un **Batch File**, il passo è breve... o quasi.

Il problema principale è rappresentato dall'editor Ed, da noi sempre citato più che altro perchè sicuramente a disposizione di tutti, essendo presente nella directory C del disco Workbench.

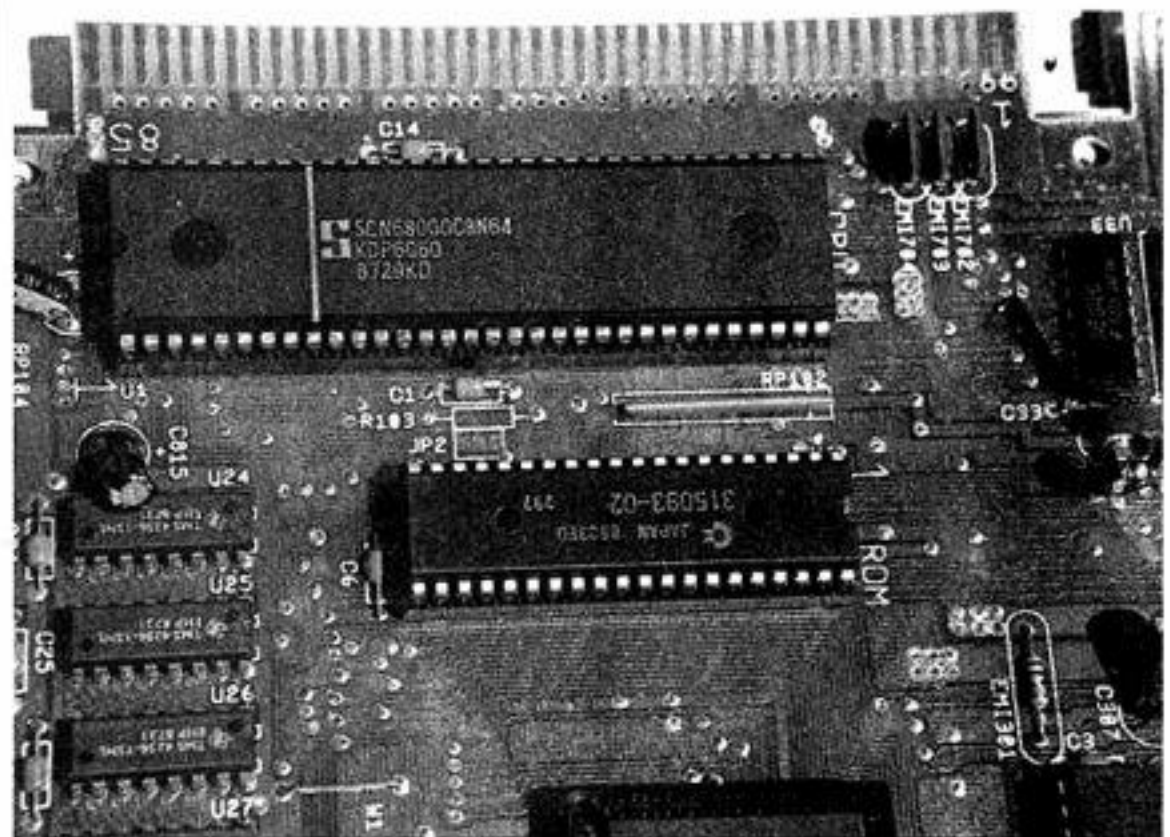
Volendo inserire il comando **Echo** appena visto in un batch di nome **Test**, memorizzato in **Ram Disk**, è, per esempio, necessario prima un comando **Cd Ram:**, quindi **Ed Test**. A questo punto, basterebbe copiare la riga Echo..., ma sorge un problema: premendo il tasto Escape, l'editor entra nel modo comandi, quindi è impossibile rappresentarlo.

Una prima soluzione viene offerta dallo stesso Dos, che interpreta come un codice Escape (Ascii 27) la sequenza di caratteri **asterisco + e** (***e**). Tutto ciò che occorre fare, è digitare la riga già vista in questa nuova veste:

Echo "*e[33;41;1m Computer Club *e[0m"

Usciti da Ed con Escape+X, si impartisca ora **Execute Ram:test**, ed il nostro mini batch produrrà l'effetto desiderato. Niente di più facile.

Ma esiste un'altra possibilità, cui Ed non può far fronte.

Fin quando si vuole inserire una sequenza Ansi nell'ambito di un comando che provvede da sè alla visualizzazione, tutto è risolvibile come appena visto, ma può sorgere la

necessità di editare un file di testo "autonomo" nel quale inserire delle sequenze di Escape.

Per non restare sul vago, poniamoci come scopo preciso quello di editare questo file (con gli ESC che ancora una volta indicano un codice di Escape, ovvero la pressione del tasto omonimo)...

ESC[32;41;1m Commodore ESC[0m
ESC[33;41;1m Computer Club ESC[0m
ESC[32;41;1m by SYSTEMS ESC[0m

...cui daremo ancora Test come nome, ed il cui scopo è quello di essere visualizzato (per esempio) mediante un comando **Type Test**, esibendo una certa eleganza grafica.

Stavolta, sostituire ad Escape la sequenza "*e" non servirebbe a niente: Type si limiterebbe a mostrare il file esattamente come lo si vede nelle righe che precedono, con i caratteri "*e" ben visibili al posto degli Esc.

Soluzioni? Più d'una, e tutte che possono tranquillamente essere applicate anche ai casi precedenti.

Prima tra tutte, quella di... mandare al diavolo Ed.

Tanto i batch file, quanto un eventuale file-testo come quello appena proposto, richiedono, infatti, solo che siano memorizzati, o se preferite "salvati" sotto forma di files Ascii.

ED lo fa, ma anche molti altri editor, sicuramente più comodi del vetusto residuato che mamma Commodore continua ad inserire nei dischi di sistema di Amiga.

Una volta deciso di lasciare Ed, la scelta dovrebbe ricadere su **Editor Ascii** che consentano la rappresentazione del carattere Escape, ma la scelta non manca. Qualche nome: **Az**, **Supered** e **TxtEd** per restare nell'ambito del pubblico dominio, o ancora **Cygnus Ed Professional**.

Con tutti questi è sufficiente copiare il file prima proposto adoperando direttamente il tasto Escape ove necessario, e salvare il file (che sarà in Ascii). Provvederà poi un nostro Type Nomefile a mostrarlo in tutto il suo splendore.

Ancora meglio, si può ricorrere ai cosiddetti **Ansi Editor**, che consentono di inserire in modo intuitivo (senza doverne digitare fisicamente i codici) le caratteristiche desiderate. Uno per tutti citiamo **Ansi-Paint**, del mitico Steve Tibbett, nel quale basta selezionare, da menu, i colori e lo stile dei caratteri, e digitare il testo desiderato: pensa lui a trasformare il file ascii prodotto in modo che contenga le sequenze di Escape necessarie.

Se poi si vuole proprio restare sull'artigianale, si può sempre adoperare la risorsa più vicina che c'è: la Shell stessa!

In fondo, abbiamo già visto che i caratteri Escape li accetta senza problemi, anzi li visualizza pure. Se a questo aggiungiamo che la console attiva (rappresentata dal carattere "*") può essere diretta altrove, per esempio verso un file, si sperimenti questa procedura:

1) Impartire da Shell il comando...

Copy * to ram:test

Il cursore si porterà sulla riga successiva, priva del prompt.

2) Copiare il testo del file prima riportato, adoperando il tasto Escape quando specificato. Alla fine di ogni riga, si pressi il tasto Return, badando che, dopo, non è più possibile correggere la riga digitata.

3) Ultimata la copia, si pressi **Ctrl+** (barra inversa).

In Ram Disk sarà ora presente un file Ascii di nome Test, a meno che non si sia adoperato un altro Path (per esempio **Copy * to Df1:test**).

Si impartisca, finalmente, **Type Ram:test**, e, se non si sono commessi errori, il file verrà visualizzato in una veste grafica di tutto rispetto.

Infine, non resta che citare un'ultima possibilità (escludendo l'orrido Edit): utilizzare ugualmente Ed, e provvedere in un secondo tempo ad un "replace".

Ovvero: digitare con Ed il file testo di cui sopra, sostituendo al carattere Escape un qualunque simbolo di solito poco adoperato, come per esempio la barra verticale (|), la back slash (\), o qualunque altro di vostra preferenza, purchè non sia presente nel testo da visualizzare.

Una volta salvato il file con Escape+X, è poi necessario un programma che legga il file, ne controlli i caratteri uno per uno, e sostituisca con un **Chr$ (27)** ogni ricorrenza del carattere adoperato per sostituire l'escape.

Un simile programma, che sfrutta un basic abbastanza elementare, potrebbe essere

```
INPUT "File da trattare"; file$
PRINT : file2$ = file$ + ".ANS"
caratt:
INPUT "Carattere da sostituire"; car$
PRINT: IF LEN (car$) > 1 THEN caratt
PRINT "Sostituisco" car$ " con Escape"
PRINT "Sei sicuro? (S / N)":PRINT
loop:
a$=INKEY$: a$ = UCASE$ (a$)
IF a$ <> "S" AND a$ <> "N" THEN loop
IF a$ = "N" THEN END
PRINT: PRINT "ATTENDI...": PRINT
OPEN "i", #1, file$: OPEN "o", #2, file2$
WHILE NOT EOF(1)
 a$ = INPUT$ (1, 1)
 IF a$ = car$ THEN a$ = CHR$ (27)
 PRINT #2, a$;
WEND
PRINT "FATTO": CLOSE: PRINT: END
```

Programma per manipolare il carattere ESC

strutturato come indicato nel riquadro di queste pagine.

Mandato in esecuzione, chiede in Input il nome del file (eventualmente completo di percorso) da elaborare, ed il carattere che si è adoperato al posto dell'Escape. Quindi, senza manomettere il file originale, crea nella stessa sua directory un nuovo file con lo stesso nome e suffisso "**.Ans**", pronto per essere adoperato con Type. La velocità non è il suo forte, ma per ottenere qualcosa di più sarebbe indispensabile ricorrere alle librerie di sistema, rendendolo poco chiaro ai meno esperti. Le scelte sul come operare, come si sarà notato, non sono davvero poche.

Notepad, ci sei?

Ho memorizzato su disco alcuni files generati con Notepad.
Successivamente ho deciso di cambiare nome al disco, quindi vi ho memorizzato altri files sempre creati col Notepad.
Il problema è che riesco a caricare solo questi ultimi, mentre con i precedenti appare la richieta di inserire un disco col vecchio nome, costringendomi a continui Rename del disco.
Sapreste dirmi perchè?
(Ettore Saporito - Milano)

Anzitutto, è necessario precisare che il problema non si pone se il caricamento di un testo viene effettuato **dopo** il lancio di Notepad. In questo caso, è sufficiente scegliere l'opzione **Open**, e specificare quindi il nome del file, comprendendo l'eventuale percorso.

Se il file si trova, per esempio, nella directory principale di un disco rinominato **Work**, basterà inserire...

Work:nomefile

...e tutto procederà per il verso giusto.

L'inconveniente lamentato dal lettore può invece essere riscontrato quando si accede al Notepad biclickando direttamente l'icona del documento redatto.

La causa non è da imputarsi alla posizione del file, ma piuttosto a quella del Notepad. Si provi a clickare una sola volta sull'icona di un documento, selezionando, poi, l'opzione **Info** dal menu Workbench.

Nel riquadro **Default Tool** della finestra Info, si potrà leggere il percorso necessario per attivare il Notepad, comprendente, per esteso, il nome del disco e di eventuali subdirectory.

Modificando il nome del disco ove è memorizzato Notepad, l'informazione contenuta nell'icona dei files-documento rimane tuttavia inalterata, per cui, biclickando su di essi, il sistema verrà tratto in inganno.

Le soluzioni possibili sono più d'una, prima tra tutte quella di memorizzare i documenti in un disco diverso da quello contenente il Notepad.

Rinominando il disco dei testi, il percorso memorizzato nella loro icona (che, ribadiamo, si riferisce al floppy che contiene Notepad) rimarrà ugualmente valido, e tutto funzionerà correttamente... a meno che non si rinomini anche il disco contenente il text editor.

In alternativa, dopo aver rinominato il disco contenente sia Notepad che i files, è possibile accedere alla Info dell'icona dei testi seguendo la procedura prima descritta, poi clickare una volta nel riquadro Default Tool, e modificare manualmente il nome del disco quivi inserito.

In questo caso, come ovvio, la stessa operazione andrebbe ripetuta per tutte le icone-testo presenti nel floppy.

Infine la più ovvia, che aggira il problema alla base: per attivare il word processor (beh, se proprio lo si vuole considerare tale...) non ricorrere al doppio click sul documento, ma agire **una sola volta** sull'icona del Notepad, premere lo **Shift** e, mantenendolo abbassato, **biclickare** sull'icona del file di testo.

In tal modo si aprirà il documento **indipendentemente** dal percorso di ricerca.

In confidenza, però, la soluzione più opportuna è anche la più semplice:

lasciar perdere Notepad, ed usare un "buon" elaboratore di testi.

Vietato l'accesso

Nel disco Extras 1.3, directory BasicDemos, ci sono le librerie tipo "graphics.bmap" eccetera. Dopo averci clickato sopra, però, non succede niente, eccetto una scritta che appare sullo schermo. Perchè?
(Alfredo Mazzuoli - Cinisello Balsamo)

Perchè non si tratta di "programmi", ma di files che eventualmente possono essere adoperati da altri programmi scritti in basic. Senza dilungarci su un argomento più volte trattato, e che viene frequentemente affrontato negli articoli che riguardano la programmazione, chiariamo al lettore, evidentemente alle primissime armi, (ed a tutti quelli come lui) che non tutto ciò che ha un'icona può essere mandato in esecuzione.

La scritta che appare sulla barra del workbench quando si clicka su questi files con suffisso **.bmap** è "The icon have no default tool", che a grandi linee significa: all'icona non è collegato alcun nome di programma che ne fa uso.

In questo particolare caso, l'icona serve solo per far vedere che questi files ci sono, anche se non sono direttamente eseguibili.

Se vi si clicka sopra una sola volta, e poi si sceglie "info" dal menu Workbench, si potrà constatare come indichino un "tipo" **Project**, cioè, per semplificare, qualcosa prodotto da un altro programma o che serve ad un altro programma.

A quale di essi, è specificato nel riquadro **Default Tool**, che nei files in questione risulta vuoto (ecco il perchè della scritta).

Nella stessa directory, tra l'altro, sono presenti altre ico-

ne, forse più note, che caratterizzano i programmi basic.

Andando a guardare la loro "Info", si noterà che anche queste sono di tipo "project", ma nel riquadro Default Tool vi si troverà scritto "**:Amigabasic**". In altre parole, quando vi si clicka sopra due volte, viene **prima** caricato il "Tool" Amigabasic, **poi** il "project" legato all'icona (il testo del programma); quest'ultimo, senza il primo, non avrebbe alcun senso: a che cosa servirebbe un listato basic senza l'interprete?

Quanto alle "librerie", che poi in senso stretto tali non sono, pur essendo lodevole la volontà di approfondire (l'occasione, comunque, su C.C.C. non manca certo), si consiglia di assimilare dapprima quantomeno i "rudimenti" del basic, nonchè quelli più generici dell'uso di Amiga e del suo Dos.

Dopo, tutto si farà più facile.

Il criticatutto

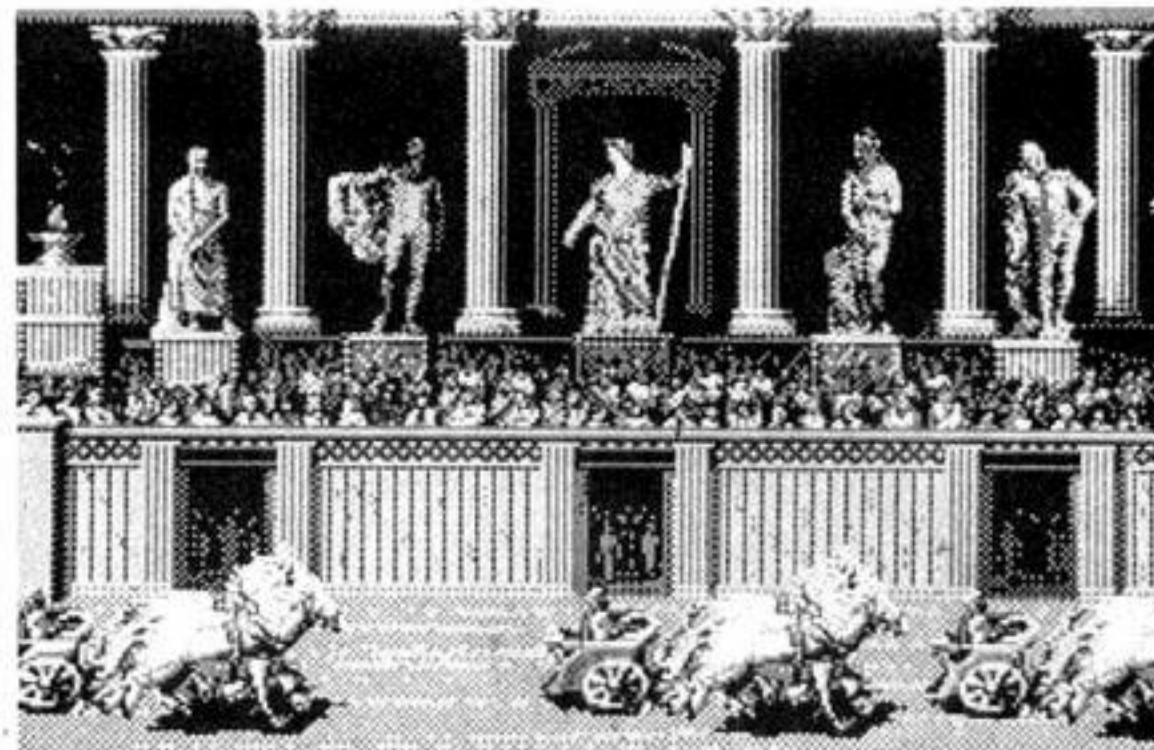
Vorrei far notare una macro-contraddizione: sul n. 72, rubrica Postamiga, un lettore viene liquidato affermando che non si può certo spiegare come pirateggiare un programma, chiamando in causa la Commodore Italia e la Software Limited. Ma a pag. 14 del numero 71 di CCC, il recensore spiega proprio quali copiatori utilizzare e addirittura come fare (sul C/64, n.d.r.). Detto questo, vorrei chiedervi se vale la pena acquistare una espansione di memoria che porterebbe la Ram del mio A-500 a 2 Megabyte, o ci si può accontentare di una da 512 Kb.

(Mauro Bossetti - Milano)

La lettera del nostro lettore, in effetti, era molto più lunga e circostanziata, ma siamo stati costretti a sintetizzare per ovvi motivi di spazio.

Quanto alla contraddizione, bisogna premettere che nell'ambito della rivista vige la più ampia democrazia: il megadirettore decide, gli altri si adeguano.

Questo però non toglie che Egli, nella sua enorme bontà, non lasci (talvolta) liberi i collaboratori di esprimersi (quasi) come meglio credono.

Certo se qualcuno di loro (noi) scrivesse che Amiga è un computer ad 8 bit non la lascerebbe passare una prima volta, si "adombrerebbe" una seconda... alla terza non è mai sopravissuto chi potesse raccontarla. Ora, pur senza considerare che il nunero 71 corrisponde ad un momento un po' particolare nell'evoluzione della rivista (vedi Suo editoriale del numero successivo), i riferimenti citati sono frutto di due "penne" (word processor) diverse. Uno dei due (ehm) in particolare, su Postamiga, non poneva una questione morale

(che potrebbe anche essere valida), ma semplicemente non voleva... rischiare il tribunale, trattandosi tra l'altro di prodotti professionali. Contraddizioni simili, in fondo, non fanno male a nessuno: rientrano nella categoria dell'"umano".

Quelle tecniche, semmai, sarebbero realmente da stroncare.

Prima, però, occorrerebbe trovarle...

Quanto alla domanda, anche questa può essere "liquidata" in breve: Amiga è una cara, simpatica bestiolina affamata di Ram. Più ne ha, meglio è.

Certo può accontentarsi anche dei suoi 512 Kappa in dotazione, ma, se si è in grado di spendere qualcosina in più senza sacrificare altri elementi importanti (seconda casa, barca da 22 tonnellate, secondo drive, stampante, ecc.), con due Megabyte si può stare davvero molto comodi...

Maledetta stampante!

Come accidentaccio devo fare per stampare su carta un testo così come lo vedo sullo schermo? Uso il Prowrite, e le opzioni Nlq e Draft per la stampa normale funzionano benissimo, ma il modo "Standard" (per la grafica) non ne vuole sapere! Non rispondetemi di consultare meglio i manuali...

(Tonino Giorgi - Ascoli Piceno)

Il problema, come peraltro "sospettato" nel resto della lettera, è quasi certamente da collegarsi alla scelta del giusto **Driver** tramite le **Preferences**, dando per scontato che Prowrite sia adoperato correttamente (sicuro, vero?).

Per una Commodore con emulazione Epson, nella directory Devs/Printers del disco Extras 1.3 è presente un file (driver) Epson X [CBM_MPS - 1250] che dovrebbe andar bene. Si tratta di **copiarlo** nell'omonima directory del disco contenente il **Prowrite**, o comunque di quello adoperato per il **boot** di Amiga, e selezionarla poi da Preferences.

Salvata la configurazione delle Preferences, l'opzione "Standard" del Prowrite (che dalla versione 3.0 è diventata "Normal") dovrebbe riprodurre su carta anche font particolari o immagini grafiche inserite nel testo.

Quanto alla postilla: un'attenta consultazione della manualistica, tanto della stampante che del word processor, non può che far bene...

CD

CD
Consente di conoscere la directory attuale, o di posizionarsi in una certa directory. Cd sta in fatti per **C**urrent **D**irectory, ed assume particolare importanza nella gestione dei files, considerando la peculiare configurazione "ad albero" consentita dal dos di Amiga.
Come noto, infatti, anche escludendo l'uso di hard disk, in un floppy possono essere immagazzinati centinaia di files. L'uso di directory e subdirectory permette di dare un ordine logico ai dati, in modo che risultino più facilmente raggiungibili: Cd, consente in pratica di passare agilmente da una all'altra. Conoscere in qualunque momento la directory nella quale "ci si trova" è inoltre di estrema importanza, in quanto tutte le attività di lettura, memorizzazione o spostamento di files saranno rivolte esclusivamente a questa directory, a meno che non venga espressamente specificato il contrario.

CD. Impartito da solo, si limita a visualizzare il nome della directory corrente. Se, per esempio, dopo aver lanciato il sistema col disco Workbench, si apre una finestra Shell e si adopera CD, sul video verrà specificato "**Workbench1.3:**" con i due punti finali che stanno ad indicare che si tratta della directory principale (chiamata Root Directory) del disco.

CD nome / :

: **(simbolo due punti)**; se adoperato senza altri parametri, consente di riportarsi nella directory principale, indipendentemente dalla posizione attuale. Quindi con...
Cd :
...pur trovandosi nell'ultimo "livello" (**d**) di una ipotetica ramificazione **Ram:a/b/c/d**, la directory corrente diventerebbe Ram:, o Ram Disk che dir si voglia.

Nome. Aggiungendo come parametro di Cd il nome di una directory, questa diventa la directory corrente.
Oltre che ad una directory, si può fare riferimento anche ad un device logico o fisico.
Quindi, per fare un esempio, con **Cd miadir** si imposterà Miadir come directory corrente, purchè questa esista già, ovvero la si sia preventivamente creata con Makedir Miadir.
Potrà sembrare un esempio banale, tuttavia occore tener presente alcune considerazioni, riportate nel riquadro a parte.

/ **(barra obliqua)**; oltre che per suddividere il percorso in subdirectory, se adoperata da sola (senza il parametro **Nome**!) serve anche per "risalire" ad una directory "superiore". Continuando ad adoperare, come esempio, la ramificazione creata nella descrizione del parametro Nome (vedi paragrafo specifico), ci si porti nella directory **B** con **Cd Ram:a/b**.
Ora, volendo impostare come *corrente* la directory **A**, un comando **Cd A**, nonostante la contiguità, **non** sortirebbe alcun effetto. In questo caso, si adopera un semplice **Cd** /. Lo si provi; poi, per verifica, si impartisca **Cd** (da solo): se tutto è in regola, sul video dovrebbe apparire "*Ram Disk:a* ", ovvero l'attuale directory. In presenza di annidamenti più profondi, possono anche essere adoperate più barre oblique. Per esempio, in una ramificazione **Ram:A/B/C/D**, per risalire dalla D alla B andrebbe adoperata la forma **Cd** //.
Nulla vieta, naturalmente, che si adoperi la più esplicita (ma più scomoda) forma **Cd Ram:a/b**, ovvero specificando l'intero percorso a partire dalla root.

Attenzione a...

Non confondere l'uso dei due punti **prima** di un path e quello posposto al nome di unità periferiche fisiche o logiche. Mentre è univoco il significato di unità come **Df1:**, **Ram:**, eccetera, diverso è il significato (per esempio) di **:C** e **C:**.

Nel primo caso, si intende la directory **C** del disco che si sta esaminando, che può quindi trovarsi in qualunque unità (Ram, df1:, hard disk, eccetera).

Con **C:**, invece, si intende solo ed unicamente la directory nella quale il sistema cerca i comandi eventualmente digitati da Shell.
Questa, dopo lo start, corrisponde alla directory **C** del disco di boot, ma può essere riassegnata dal comando Assign (si adoperi solo Assign per vedere quali sono le assegnazioni vigenti).

Nome: dissertazioni

Più che le parole, qualche intervento sulla tastiera renderà tutto ben chiaro. Si lanci il sistema con il disco Workbench1.3, poi si entri in ambiente **Shell** biclickando sulla sua icona.
Ora, si digiti...
Cd Ram:
Con questo comando, si sarà impostata, come directory corrente, la Root directory della **Ram Disk**. Il che, tradotto in pratica, significa che se, per esempio, ora digitiamo un comando **List** privo di parametri, questo tenterà di mostrare il contenuto della Ram disk, ovvero della directory corrente. Analogamente, se si digita il nome di un programma, questo verrà ricercato **prima** in Ram Disk, poi, se non è presente, nella directory **C** del disco di boot (nel nostro caso, il Workbench), ed infine nelle **eventuali** altre directory inserite nel **Path** (percorso di ricerca) impostato nella startup-sequence (si veda descrizione del comando Path sul n. **75**).
Potremo quindi dire che, con Cd Ram:, abbiamo anche aggiunto la directory principale della Ram Disk nel Path di sistema.
Ora si provi ad impartire...
Makedir A
...(e return), quindi **Cd A**. Come mostrato dal prompt della shell, ci troviamo ora all'interno della nuova directory di nome "**A**" appena creata. Si ripeta l'operazione creando stavolta una subdirectory di nome B (**Makedir B**): questa, si troverà annidata all'interno della directory "A". Ora risaliamo nella root impartendo di nuovo **cd Ram:**.
Supponiamo ora di volerci spostare nella directory B. Si provi quindi a digitare **Cd B**. In risposta, Amiga risponderà **Can't find B**, ovvero che non trova questa directory. Ecco dunque uno dei punti cui prestare sempre attenzione: per muoversi da una directory all'altra, è necessario non solo che questa esista già, ma anche che sia "visibile" **da** quella attuale. Nel nostro caso (lo si provi con un **Dir** oppure **List**) solo **A** risulta accessibile **direttamente**. In questo caso, è necessario precisare il percorso necessario per raggiungere B, ovvero **CD A/B**, con la barra obliqua posta a separare i nomi delle subdirectory, scritti senza spazi. Fanno eccezione a questa regola tutti i nomi di periferica fisica o logica, nonchè i nomi di disco, ovvero le loro directory principali. Detto più terra-terra, tutti quegli elementi che finiscono con il simbolo di doppio punto(:). Si provi, per esempio, dalla directory B nella quale ci si trova, a digitare **Cd Df0:**. Immediatamente la directory attuale diventerà la root del disco Workbench. Proprio per questa caratteristica, è spesso più comodo adoperare i nomi di device logico piuttosto che quello di directory, quando ovviamente possibile.
Continuando a smanettare sulla tastiera, si impartisca ora (ci si dovrebbe trovare con Workbench1.3 specificato nel prompt) **Cd Ram:a/b**. Con una sola istruzione, si è passati ad una subdirectory in Ram (occhio ai due punti!).
Ora, cerchiamo di spostarci nella directory **C** di sistema. Potremmo adoperare **Cd Workbench1.3:C**, ma anche, molto più comodamente, **Cd C:** (con i due punti finali!). In tal modo, anche se (per esempio) in un secondo drive fosse presente un'altra directory **C** (o in Ram stessa), verrebbe in ogni caso presa in considerazione la C del disco adoperato per il boot, o quella eventualmente installata con **Assign** (si veda Amigafacile n. **75**).

Doppio punto: dissertazioni

Molto utile risulta la possibilità di preporre questo simbolo ad un eventuale percorso, nel caso si presenti la necessità di risalire l'intero path per poi "ridiscenderlo" in altre direzioni.
Per un riscontro pratico, si impartisca, da Shell, l'istruzione...
Cd Sys:
con **Sys:** a rappresentare il device logico che indica la root directory del disco di sistema, quello adoperato per il lancio, nel nostro esempio Workbench1.3.
Si impartisca ora **Dir** per dare un'occhiata al contenuto del disco, parte del quale risulterà suddiviso in un certo numero di directory.
Proviamo ad entrare in una di esse, per esempio Devs, con **Cd Devs**. Un altro Dir, mostrerà la presenza di ulteriori subdirectory. Continuiamo dunque il nostro "viaggio" (verso l'ignoto?) con **Cd Printers**. Il prompt di Shell ci informerà esattamente di tutto il percorso fin qui impostato, esattamente come se si fosse impartito **Cd** da solo.
E se volessimo, a questo punto, portarci direttamente nella directory **System** del dischetto? Si potrebbe impostare, per esempio:
Cd Workbench1.3:system
...o anche...
Cd //system
(la doppia barra obliqua fa risalire di due livelli, praticamente fino alla Root). Ma anche, in base a quanto prima affermato,
Cd:system.
I due punti ci "fionderanno" nella directory principale, dalla quale è poi impostabile System, "visibile" solo da quella posizione.

ED

ED

Ed è un editor Ascii, cioè un programma che consente di elaborare dei testi che vengono poi memorizzati in accordo con lo standard Ascii.
Pur non brillando per le sue prestazioni, molto poco "amighevoli", è tuttavia uno strumento che può risultare utile nella digitazione di **brevi** listati o batch files, piuttosto che nella elaborazione di testi vera e propria, a meno che questa non sia finalizzata ad una rappresentazione su schermo.
Il motivo è presto detto: Ed inserisce automaticamente alla fine di ogni riga un **EOL** (**E**nd **O**f **L**ine), nella fattispecie un codice Ascii 10, che in una normale visualizzazione (tanto su video che su carta) produce un semplice ritorno a capo, ma che viene anche interpretato come fine di una riga di comando dal Dos di Amiga, dagli interpreti basic, dai compilatori, dagli assembler, e chi più ne ha più ne metta.
Nei veri word processor, invece, l'EOL (se voluto) va inserito manualmente con la pressione de tasto Return (o Enter), in questi tipi di programma adottato più come fine paragrafo che come fine linea.

NOTA BENE

Le parole chiave di ogni comando sono rappresentate in maiuscolo e vanno (eventualmente) adoperate così come sono.
In minuscolo, sono invece riprodotti i parametri che vanno ridefiniti dall'utente.

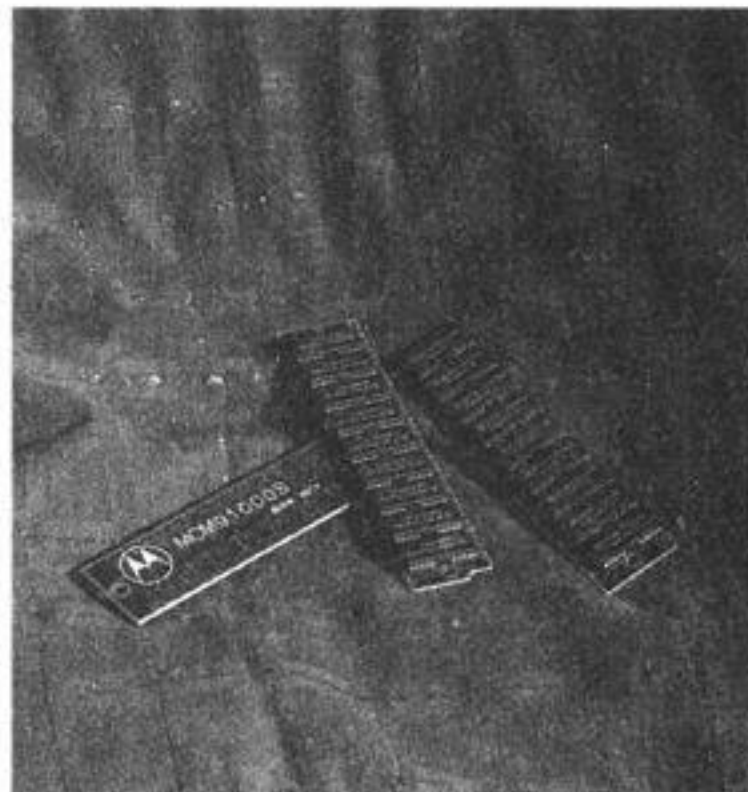
From, Size: Keyword opzionali, eliminabili senza alcuna conseguenza.

ED FROM nomefile SIZE bytes

Bytes. Facoltativo. Dopo il nome del file, può essere specificata la dimensione in byte dell'area di lavoro di Ed. Per default, questo parametro è impostato a **40.000**, più che sufficiente per la maggior parte delle applicazioni, considerando anche il fatto che, per l'editing di file molto corposi, è decisamente più salutare rivolgersi ad editor più comodi. Qualora si volesse elaborare un file eccedente le dimensioni di 40000 byte, basta inserire il valore desiderato, per esempio con **Ed miofile 90000**

Nomefile. E' il nome del file da editare, eventualmente completo di percorso (per esempio, **Ram:nomefile**). Se un file con tale nome è già presente, questo viene caricato nell'editor e può così essere modificato; in caso contrario ne viene creato uno.
Il file, se già esistente, deve **obbligatoriamente** essere in Ascii, ovvero composto di solo testo, altrimenti Ed ne rifiuterà l'editing con la laconica comunicazione **File contains binary**.
Quando si modifica un file preesistente, Ed cerca di salvaguardare l'originale non cancellandolo del tutto, ma salvandone una copia nel device logico **T:**, ovvero in una directory **T** che, nel caso del workbench 1.3, viene creata in Ram disk ed assegnata come device logico dalla startup-sequence del dischetto.
Opportuno comunque, anche se si adopera un proprio disco personalizzato, assegnare T: ad una directory in Ram: o sullo stesso disco, a seconda dell'importanza del testo che si pensa di elaborare.
La copia originale del file modificato assume sempre il nome **Ed-Backup**.

Ne abbiamo già parlato

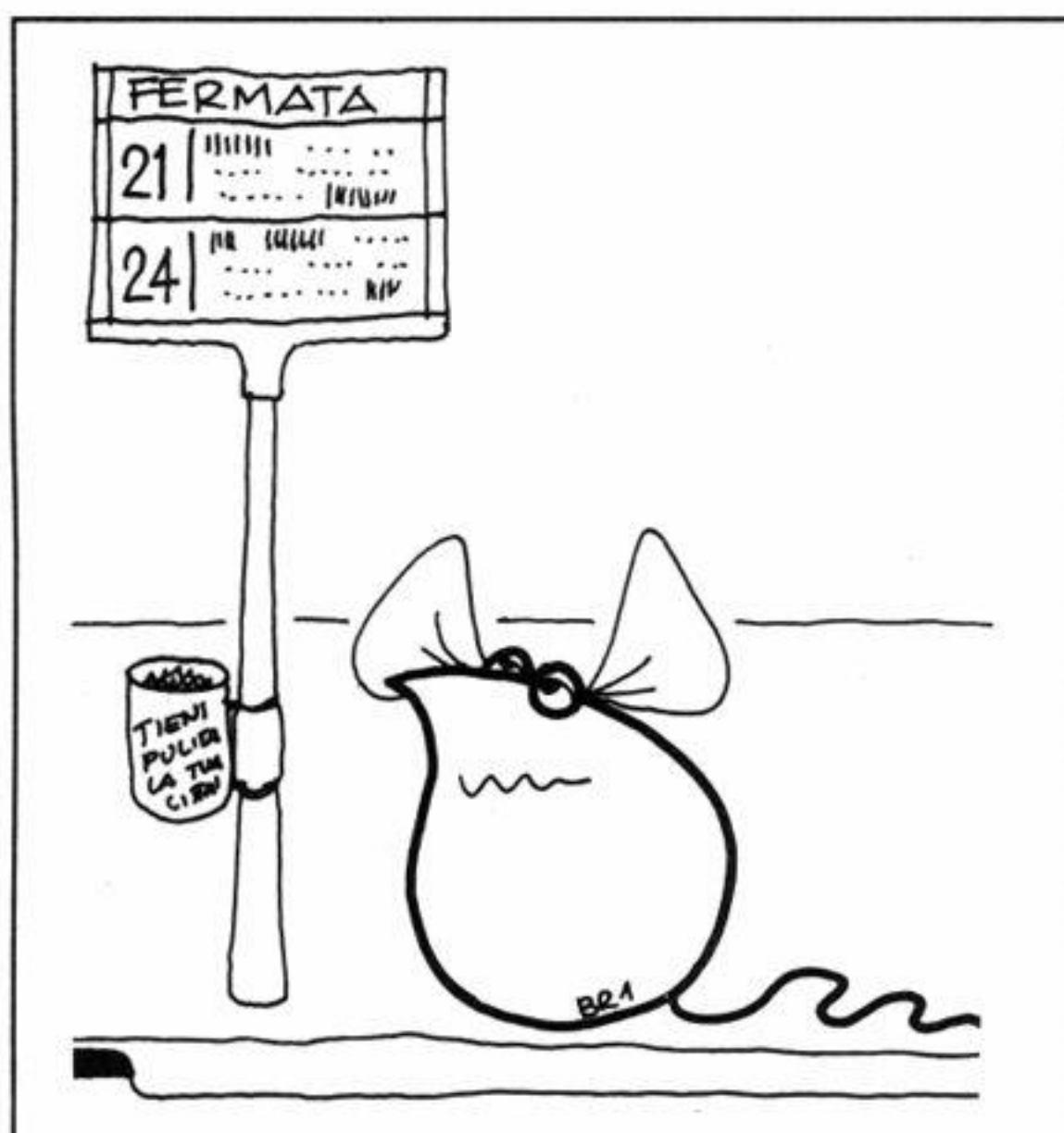
N.75	N.76	N.77
Assign	Caratteri	Execute
Copy	Speciali	Direttive
Date	Delete	Batch
Dir	Format	If
Install	Protect	Skip...lab
Path	Rename	Quit
Search		
Sort		

ED, attenzione a...

Se, per un motivo qualunque, si carica con Ed un testo Ascii elaborato con un normale word processor, si presti **molta** attenzione prima di salvarlo.
Ed, infatti, come già rimarcato, considera il Line Feed (**Ascii 10**) come un **fine linea**, ed una linea può contenere al massimo **256** caratteri.
La maggior parte dei text editor, invece, inseriscono il Line Feed come **fine paragrafo**, per cui si può avere una sequenza di caratteri di lunghezza anche molto maggiore di 256, senza la presenza di alcun line feed.
Ebbene, quando un testo siffatto viene caricato in Ed, le singole linee vengono **troncate** quando superano le 256 colonne, con relativa perdita del contenuto eccedente.
Le linee che superano le 80 colonne (oppure 60, a seconda della risoluzione di schermo adottata), non sono immediatamente visibili ma necessitano di uno scroll orizzontale (si veda tabella dei comandi), per cui è facile essere tratti in inganno, e non accorgersi di quanto è successon anche se, per la verità, una tale evenienza è segnalata da Ed subito dopo il caricamento.
Risalvando il tutto, il testo potrebbe quindi risultare mutilato. Tra l'altro, è bene non fidarsi troppo del backup automatico di Ed, molto spesso imprevedibile, che può anche non venire effettuato a seconda della directory in cui ci si trova, dagli assegnamenti, eccetera.
Un consiglio: evitare tassativamente questa possibilità, mentre il contrario (da Ed ad altri processori di testo) è sempre consentito.

Modo esteso

Nella tabella sottostante è inclusa una serie di comandi implementabile mediante l'uso del tasto **Esc**ape, che precede uno o più caratteri. I simboli illustrati, sono quindi da inserire **dopo** la pressione dell'Escape, e vanno fatti seguire dalla pressione del tasto Return.
La stessa funzione del tasto Escape può essere svolta da **Ctrl+[**, mentre con **Ctrl+G** può essere ripetuto l'ultimo comando eseguito, senza doverne di nuovo digitare l'intera sequenza.
Si tenga anche presente che dopo la pressione dell'Escape si possono digitare più istruzioni nella stessa linea di comando (che appare in fondo allo schermo), purchè separati da un simbolo di punto e virgola(;). Inoltre, preponendo ad un comando un numero (per esempio **5CR**), questo verrà eseguito tante volte quante specificate dal numero in questione.
Molti dei comandi, peraltro, risultano superflui, in quanto impartibili più comodamente tramite il "modo diretto", o sfruttando i comuni tasti di Amiga.

Cl	Muove cursore a sinistra di un carattere.
Cr	Muove cursore a destra di un carattere.
Cs	Muove cursore ad inizio riga.
Ce	Muove cursore a fine riga.
P	Muove cursore ad inizio della riga precedente.
N	Muove cursore ad inizio della riga successiva.
T	Cursore all'inizio del file.
B	Cursore alla fine del file.
Mxx	Muove cursore alla riga numero xx.
Dc	Cancella carattere sotto il cursore.
D	Cancella l'intera riga ove si trova il cursore.
S	Spezza la riga nel punto dove si trova il cursore (stessa funzione di Return).
J	Unifica la riga corrente con quella che segue.
I/x xx/	Crea una nuova riga sopra l'attuale, e vi inserisce i caratteri "xxx".
A/x xx/	Crea una nuova riga sotto l'attuale, e vi inserisce i caratteri "xxx".
Slxx	Setta il margine sinistro per l'editing alla colonna xx (il testo eventualmente già presente non viene aggiornato alla nuova posizione).
Srxx	Setta margine destro per l'editing alla colonna xx.
Ex	Estende ilmargine destro oltre la colonna impostata con SR.
Stxx	Imposta ad xx il numero di spazi percorsi dal cursore ad ogni pressione del tasto Tab.

Modo esteso (sequenze Escape)

Ricerca e sostituzione

F/xxx/	Ricerca in avanti la sequenza di caratteri "xxx" e vi posiziona il cursore.
BF/xxx/	Ricerca all'indietro la stringa "xxx" e vi posiziona il cursore.
E/xx/yy/	Cerca la ricorrenza di caratteri "xx" e la sostituisce con "yy" (una sola volta. Per ripetere usare Ctrl+G).
EQ/xx/ yy/	Stessa funzione di ricerca e sostituzione ma prima viene chiesta conferma.
UC	Imposta la ricerca senza badare alla differenza maiuscolo/minuscolo.
LC	Imposta ricerca che distingue tra maiuscolo e minuscolo.

Operazioni su blocchi di testo

BS	Setta inizio di un blocco di testo alla posizione del cursore.
BE	Fine blocco di testo alla posizione del cursore.
DB	Cancella il blocco correntemente impostato.
SB	Porta il cursore all'inizio del blocco impostato.
IB	Inserisce una copia del blocco impostato alla linea seguente.
WB/file/	Salva il blocco impostato come file-testo di nome "file".

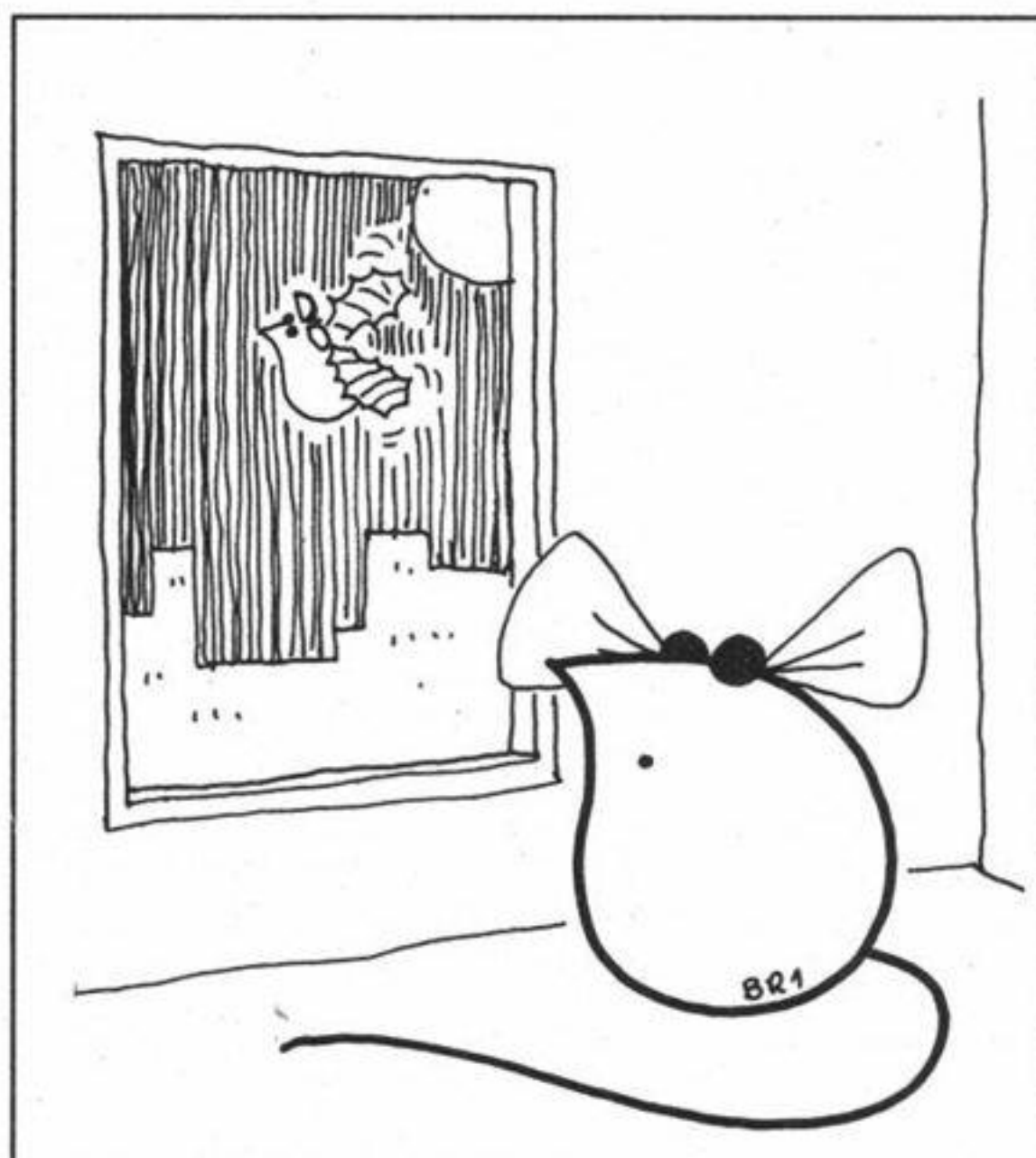
Ctrl+ T	Muove il cursore all' inizio della parola successiva.
Ctrl+ R	Muove cursore alla fine della parola precedente.
Ctrl+]	Cursore ad inizio / fine riga (in alternanza).
Ctrl+ D	Scroll di una pagina verso il basso.
Ctrl+ U	Scroll di una pagina verso l'alto.
Ctrl+ E	Cursore ad inizio/fine schermo (in alternanza).
Ctrl+ H	Cancella carattere a sinistra (come back space).
Ctrl+ O	Cancella dal cursore fino al prossimo spazio.
Ctrl+ Y	Cancella dal cursore alla fine della linea.
Ctrl+ B	Cancella tutta la linea ove alloggia il cursore.
Ctrl+ A	Crea una riga vuota sotto quella corrente.
Ctrl+ F	Inverte stato maiscolo / minuscolo dei caratteri.
Ctrl+ V	Effettua un Refresh dello schermo.

Modo Diretto
In questa categoria rientrano tutte quelle funzioni implementabili direttamente da tastiera, per lo più in associazione al tasto CTRL.
Dalla descrizione sono esclusi le più elementari funzioni di editing, vale a dire i movimenti del cursore legati ai relativi tasti, nonchè tab, Del e back space.

Il tasto ESCape

Anche in queste tre tabelle sono inclusi i comandi implementabili mediante l'uso del tasto **Esc**ape, che precede uno o più caratteri.
I simboli illustrati, sono quindi da inserire **dopo** la pressione dell'Escape, e vanno fatti seguire dalla pressione del tasto Return.

Varie

SH	Mostra alcune informazioni sul file e sui settaggi dell'editor.
U	Cerca di annullare l'effetto dell'ultima operazione (undo
X	Salva il file corrente con lo stesso nome e conclude la sessione.
Q	Esce dall'editor senza salvare il testo corrente.
IF/nome/	Carica il file "nome" e lo inserisce alla posizione attuale del cursore spostando eventualmente il testo sottostante.
SA/file/	Salva il testo corrente come file-testo di nome "file" (o con lo stesso nome se il parametro viene omesso)

BREAK

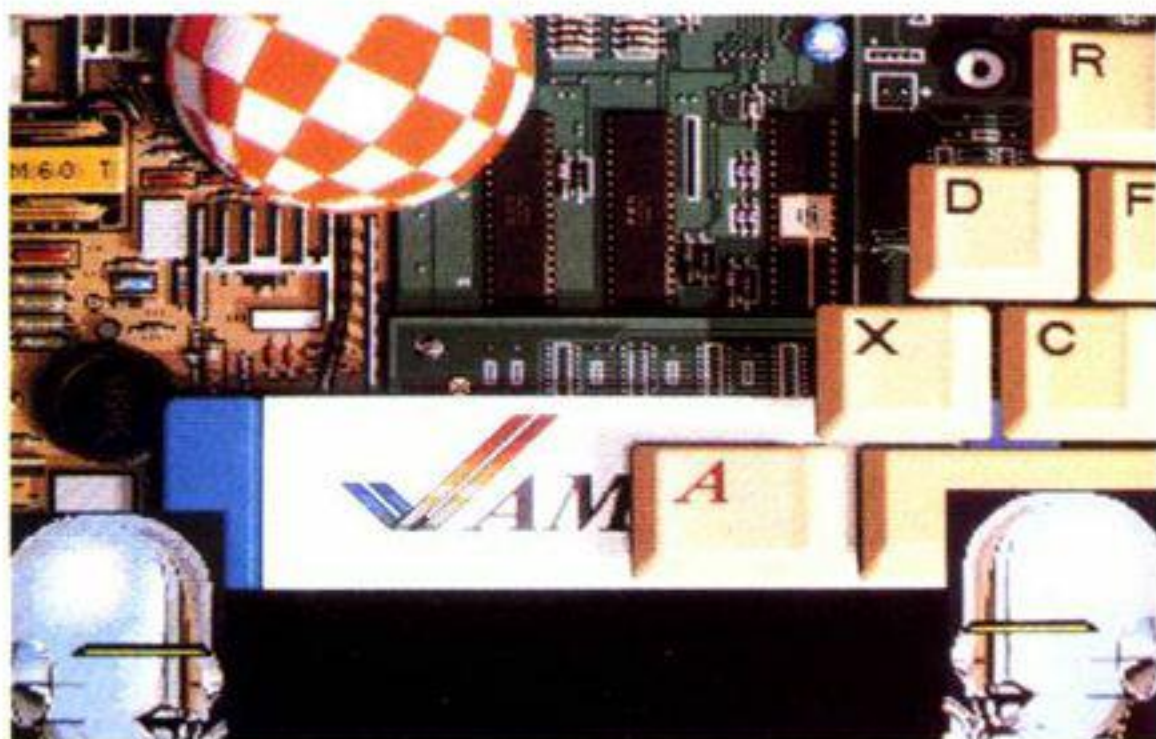
Consente di interrompere l'esecuzione di un task attivo, oppure di settarne i cosiddetti "flag di attenzione" che ne provocano l'interruzione.
Più chiaramente: un Task, grosso modo, può essere identificato da una singola unità attiva nel sistema, come ad esempio una finestra Shell.Se si lancia un qualsiasi programma da questa finestra, digitandone il solo nome, il programma impegnerà tutte le risorse di quel task, che non potrà così essere utilizzato per altri usi fino all'uscita dal programma (a meno che lo stesso non provveda a "rilasciare" il task).
Un "flag di attenzione" non è altro che una combinazione di tasti, che il sistema verifica di continuo se è stata attivata.
In caso positivo, interrompe il processo in corso.
Per default, il flag settato corrisponde al ben noto Ctrl+C, la cui azione può essere immediatamente verificata aprendo dal disco Workbench una finestra Shell, digitando il comando Dir e, mentre viene visualizzato il contenuto del disco, premendo in contemporanea i tasti Ctrl e C.

NumTask. Indica il numero di task attivo al quale vanno riferiti i parametri successivi, o, in mancanza di quetsi ultimi, il task al quale inviare l'equivalente di un **Ctrl + C**. Il numero di task, per capirci, è quello che viene indicato nel prompt di una finestra **Shell** o **Cli** (salvo manipolazioni tramite il comando Prompt). Riferirsi al riquadro specifico per maggiori informazioni.

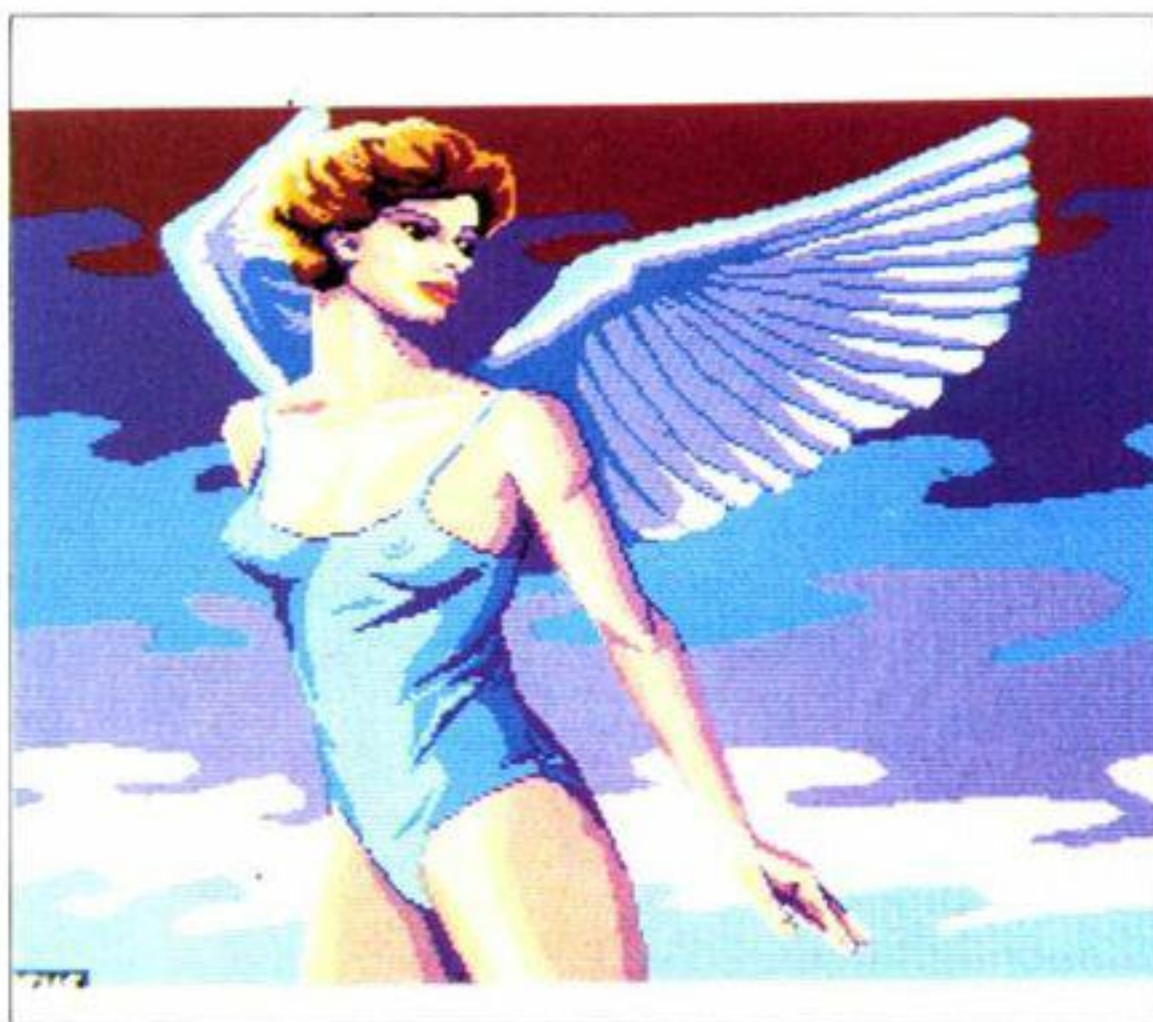
C, D, E, F. Dopo il parametro **task**, può essere inserita una o più di queste quattro lettere, per segnalare al sistema quale flag di attenzione risulta valido per quel particolare Task.

In altre parole, con "**Break 3 E**", per esempio, si provoca lo stesso effetto ottenibile clickando in una ipotetica finestra Dos n. 3, e quindi premendo **Ctrl + E**.

Se un'applicazione prevede una sua interruzione legata a questa sequenza, concluderà la sua attività.

Analogamente, si possono specificare più flag contemporaneamente; per esempio con **Break 3 E F**, o addirittura tutti (**C, D, E, F**): per quest'ultimo caso, e ' prevista l'opzione **ALL**, da far seguire al parametro NumeroTask (per esempio, **Break 3 All**).

Il numero di Task

Per non restare sul vago, si lanci Amiga con il disco Workbench1.3 e, dopo averne aperto la relativa icona, si "bi-clicki" (orribile neologismo) sull'icona Shell.
La finestra così aperta, avrà come prompt...
1.SYS:>
...dove 1 indica proprio che ci si sta "muovendo" nell'ambito del task numero 1.
Si impartisca, ora, un comando **NewShell**: verrà aperta una nuova finestra, dal prompt **2.SYS:**. Questo è il task numero 2. Per procedere con i nostri esperimenti, si riducano le dimensioni di quest'ultima finestra ad un quadrato, e lo si sposti in un angolo, badando di lasciare scoperta **anche** una porzione dello schermo Workbench.
Nel task **2** (dopo avere attivato la finestra relativa clickandovi all'interno con il mouse) si impartisca **Dir** (+ return), e, quando comincia ad apparire la lista delle directory, si prema la barra spaziatrice per arrestarne momentaneamente lo scorrimento.
Ora si attivi il **Task 1** col solito click, quindi si digiti al suo interno **Break 2** (e Return).
Nuovo click nella seconda finestra, e con il tasto Back Space si annulli l'azione della barra spaziatrice: Il comando **Dir** si interromperà con un **BREAK**, esattamente come se si fosse adoperato **Ctrl + C**.
In taluni casi il ricorso a **Break NumeroTask** può risultare decisamente più determinante che non nell'esempio appena visto. Si provi, dalla finestra numero 2, a digitare **ClockPtr** e return. Il cursore andrà a capo, ma senza riesibire il prompt. Il task è in pratica bloccato nell'esecuzione del programma, la cui azione può essere notata clickando al di fuori delle finestre Shell: il puntatore diventerà un piccolo orologio (nota: Clockptr è contenuto nella directory **Utilities** del disco Workbench, e può essere mandato direttamente in esecuzione perchè questa directory è inserita nel Path di ricerca tramite la startup - sequence).
Per arrestare l'azione di Clockptr, possiamo dunque seguire due strade: attivare la finestra 2 e premere Ctrl+C, oppure attivare la finestra 1 ed impartire Break 2: in entrambi i casi riapparirà immediatamente il prompt, e Shell sarà pronto a ricevere nuovi comandi, come in fondo si era già visto per Dir.
Si segua uno dei due metodi, quindi, ancora dalla finestra 2, ma si impartisca stavolta **Run ClockPtr**. Il prompt verrà immediatamente rilasciato, dopo un segnalazione **[CLI 3]**.
Che cos'è successo? Semplicemente, si è attivato l'orologio - puntatore in multitasking, ovvero in un suo task esclusivo (il n. **3**), che quindi non impegna la finestra shell (numero **2**).
Per verificarlo, si impartisca in una delle due finestre (o in entrambe, se lo si preferisce) il comando **Status** (e return), che ci segnalerà quali programmi stanno "girando" nei vari task: al numero 3, corrisponde proprio clockptr, pur non essendovi alcuna finestra shell con tale numero.
Volendo però arrestare l'esecuzione di Clockptr, **non** è più possibile ricorrere a Ctrl+C: non esiste una finestra dos collegata al task 3, nella quale introdurre la sequenza.
Ecco allora che Break diventa l'unica strada di uscita, soprattutto se il programma non prevede una sua autonoma scelta di interruzione.
In una delle due finestre, si impartisca dunque Break 3: dopo la sua esecuzione, anche clickando nello schermo Workbench il puntatore rimarrà saldamente ancorato alla sua tradizionale forma di freccetta.

Per una conferma definitiva, si impartisca di nuovo il comando Status: del task numero 3, non dovrebbe essere rimasta traccia.

ACCESSORI PER A-500

Amiga Synchro Expert III

La febbre del **backup** facile è una delle prime manifestazioni (quasi) morbose cui gli Amighi di fede gamofila vanno incontro.

Una giustificazione di questo tipo non è forse la più (l/r)eale, ma si sa, un motivo ai propri vizi bisogna pur trovarlo...

La soluzione più ovvia? Ricorrere ad un apparecchio hardware che "bypassi" il sistema, e consenta di copiare qualunque cosa possa essere infilata in due drive (beh, proprio qualunque cosa forse no...).

In altre parole, un'accoppiata hard/soft come Syncro Expert III.

E per hardware, si badi bene, non si intende niente di particolarmente complicato, o che necessiti di manovre a "macchina aperta". Syncro Expert è infatti costituito da un **floppy** che contiene il software, ed un accessorio che può in pratica essere definito un riduttore: due **connettori** a 23 pin, uno maschio e l'altro femmina, disposti a facce contrapposte.

Tutto ciò che occorre fare è inserire l'accessorio (a computer spento, per carità) nella porta riservata al drive esterno di Amiga, e collegare all'altra estremità il cavo del secondo drive, la cui presenza è una "conditio sine qua non" per adoperare Syncro Expert.

Sui modelli **A-2000** configurati con due drive interni, la connessione risulterà possibile solo se si dispone anche di un drive esterno.

A questo punto non rimane che accendere il computer, ed inserire il disco-programma. 3 / 4 secondi di caricamento (sembra di avere un hard disk!), e voilà: pronti all'azione.

La schermata di Syncro Expert è abbastanza consueta per chi ha già adoperato un copiatore, e non presenta alcuna difficoltà d'uso. Sulla sinistra trovano posto due tradizionali **griglie** che mostrano il procedere della copia traccia per traccia su entrambe le facciate del disco, mentre il puntatore del mouse è confinato in un riquadro sulla destra, dal quale non può essere fatto uscire.

Qui trova posto la consolle di selezione, che consente di agire su intuitive freccette per stabilire la traccia del disco dalla quale iniziare la copia, nonchè la facciata. Opzioni queste di solito impostate a zero, se si intende copiare tutto un disco.

Più succulenta, invece, la possibilità di estendere la copia al di là delle normali 80 tracce (**fino alla 82ma**), per aggirare una delle protezioni più frequenti, che consistono, appunto, nel memorizzare alcuni dati basilari in quest'area di solito ignorata dai copiatori "ortodossi".

Più in basso, un riquadro apparentemente presente solo a fini estetici, consente invece di specificare (sempre con un click del mouse) il modello Amiga sul quale Syncro Expert sta operando: 500 e 1000, oppure 2000 e 2500.

L'inizio delle operazioni di copia, che durano in media attorno ai **50 secondi**, richiede infine una manovra che sbalordirà per la sua difficoltà: occorre clickare su un ulteriore riquadro caratterizzato dalla dicitura **Go! / Stop!**.

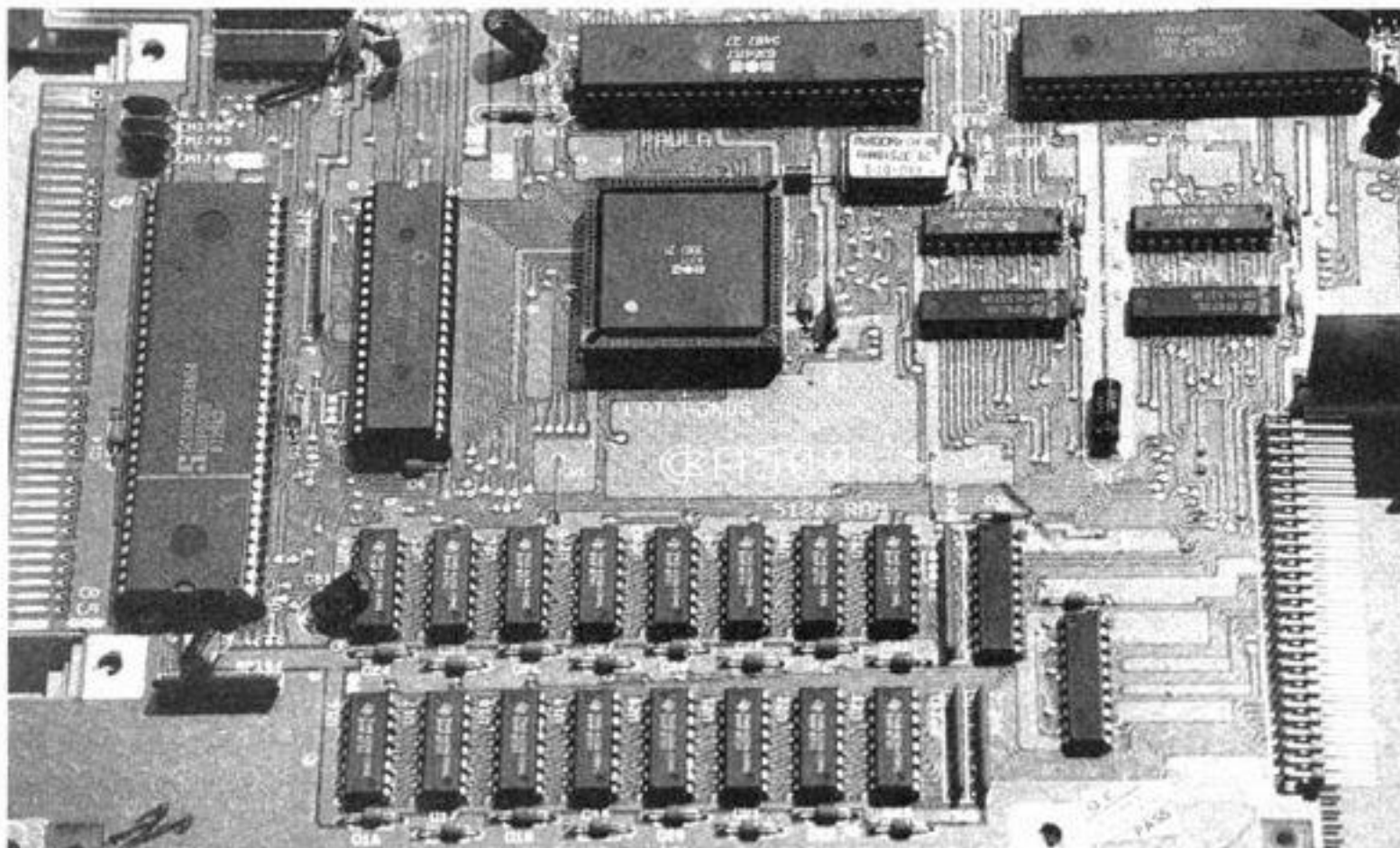
Un particolare da non sottovalutare, e del quale si è solo accennato: la copia può essere effettuata anche tra dischetti **Ms-Dos, Atari, Archimedes**, o addirittura dischi di sintetizzatori, grazie al sistema hardware adottato.

Due prodotti molto utili per i possessori di Amiga

in vendita presso
FLOPPERIA
Viale Monte Nero 31
20135 - MILANO
tel. (02) 55.18.04.84

A lavori ultimati, è teoricamente possibile uscire dal programma agendo contemporaneamente su entrabi il pulsanti del mouse, mentre il puntatore si trova all'estremità in fondo a destra dello schermo.

E' però opportuno, prima di utilizzare Amiga per i soliti scopi (ludici, è chiaro!), spegnere l'apparato, estrarre il dispositivo hardware dalla relativa porta, e reinserivi il connettore del drive.

Tutto tornerà come prima, tranne che per un particolare: non sarà più necessario trattare coi guanti quello che prima era un disco "incopiabile".

Espansione Ram 1.5 Mb

Chi trova un Amiga trova un tesoro, recita il neoproverbio del mercato mini informatico (che poi tanto mini non è), sicuramente approvato dalla sempre crescente marea di utenti del prodotto di punta di casa Commodore.

E chi trova un Amiga 500?

Alla luce degli avvenimenti più recenti, non ha più di che lamentarsi. Quello che sembrava il modello meno dotato accanto ai più "tosti" cugini della fascia superiore, sta vivendo una popolarità senza precedenti, grazie anche al proliferare di optional e upgrade riservati a questa macchina.

"Un computer nato per giocare", era l'etichetta cucita addosso al piccolo degli Amiga, e giù ad infierire: vuoi mettere al confronto con un 2000? Escludendo ogni parallelismo con i modelli 3000, che fanno discorso (e mercato) a sè, quelli che erano gli appigli formali ai quali i 2000-isti si aggrappavano, sono pian piano cominciati a cadere.

I 2000 potevano montare degli hard disk? Bene, dopo un po' è nato l'A-590, tra l'altro oggi non più unica scelta sul mercato.

Il tallone d'Achille, sempre in termini di paragone, era rappresentato però dalla memoria. Il 2000 nasce con un Megabyte già in dotazione, A-500 con solo la metà. Detto fatto, anche qui la differenza venne colmata quasi subito dalle varie espansioni tuttora in auge, che portano ad un Mega il totale di Ram disponibile.

Amiga, d'altra parte, soprattutto se non lo si usa solo per giocarci, è un computer dalle notevoli risorse: con il suo multitasking, far girare più applicazioni è la norma, ed ecco che si torna al solito problema: la memoria disponibile diventa sempre più stretta, anche con un Megabyte a disposizione.

In questa situazione, Amiga 500 sembrava dover segnare il passo, ma non poteva durare a lungo. Con una spesa più che accettabile, è ora disponibile un prodotto di tutto rispetto, introdotto sul mercato dalla **Power Computing**: una espansione di memoria che porta il 500 ad **un totale di 2 Megabyte!**

La sua sigla è **PC-502 1.5Mb**, decisamente allettante, da prendere seriamente in considerazione soprattutto ora che, grazie all'introduzione dei nuovi Agnus, si può disporre di almeno un Megabyte di cosiddetta "chip ram".

La PC-502 è costituita da due elementi hardware: il più appariscente è costituito da una basetta dalle dimensioni standard per essere accolta nell'alloggiamento posto sul fondo dello chassis di Amiga, non dotata di contenitore metallico.

Ad un primo sguardo mostra chiaramente la fattura assolutamente professionale dell'espansione, con la componentistica disposta in perfetta linea sulla superficie del circuito stampato. Spiccano in particolare la **batteria tampone** che alimenta l'**orologio** in tempo reale di cui la PC-502 dispone, e i 12 integrati da 4 x 256 Kbit, ovvero 128 Kbyte, per un "canonico" totale di 1.536.000 byte: per gli amici, un Mega e mezzo.

Da uno degli angoli della basetta, sporge poi un connettore a 6 pin destinato ad accogliere lo spinotto mono-posizione proveniente dal secondo dispositivo hardware: un controzoccolo, o adattatatore che dir si voglia, che va installato all'interno di Amiga.

Le operazioni da compiere per provvedere all'esigenza sono decisamente facili e non necessitano di alcuna attrezzatura particolare, tuttavia vanno compiute con una certa delicatezza: chi non se la sentisse di rischiare un piedino del chip **Gary** (oppure ha l'Amiga ancora in garanzia), è opportuno faccia eseguire l'operazione da persona esperta.

Si è accennato al chip Gary perchè, in pratica, occorre disinserirlo dal suo zoccolo (basta far leva delicatamente alle due estremità con un cacciavite) e innestare al suo posto l'**adattatore** fornito con la PC-502.

Naturalmente va prima aperto il contenitore del computer, rimossa la tastiera (che risulta appoggiata a dei sostegni), ed infine privata delle sue viti di connessione la schermatura metallica che copre la scheda madre di Amiga. Fatto ciò, il chip in questione è facilmente riconoscibile subito a ridosso del drive interno; di solito il suo nome (Gary) è stampigliato sul circuito stampato, ma può anche essere identificato dalla sua sigla: **5719**.

Una volta inserito il controzoccolo al suo posto, badando a far coincidere la **tacca** laterale presente nello zoccolo originale, basta infilargli sopra (delicatamente) Gary, sempre rispettando il riferimento fornito dalla tacca. Fatto.

Non resta che rimontare il tutto, badando di lasciar sporgere, dall'apertura del cassetto inferiore, il cavetto che sporge dallo zoccolo appena montato. Questo, infatti, va inserito nel connettore presente nella basetta dell'espansione, da installare infine (con i componenti rivolti verso l'interno) nel pettine posto all'interno del cassetto.

Il tutto, naturalmente, a computer spento, ed evitando di... "sbavare" sui componenti elettrici dell'espansione: la goduria è meglio riservarla al momento dell'accensione, quando sulla barra del workbench, o dopo un **Avail** da Dos, apparirà una cifra che, fino a poco tempo fa, su un 500 sembrava impossibile.

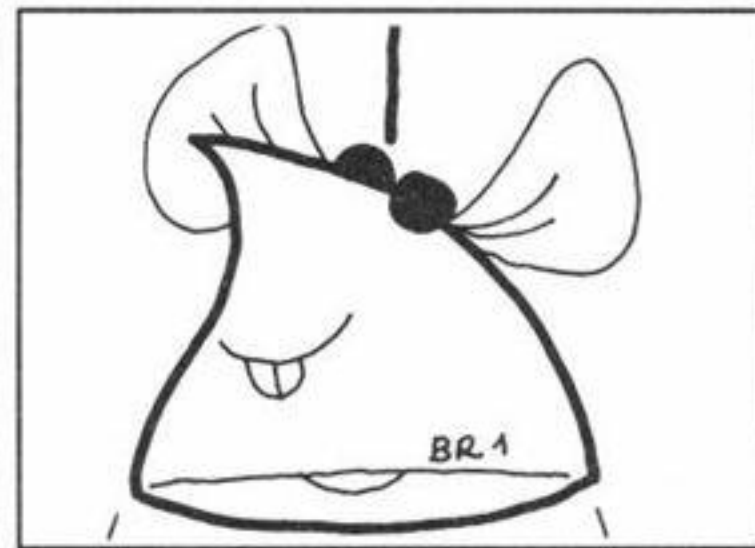
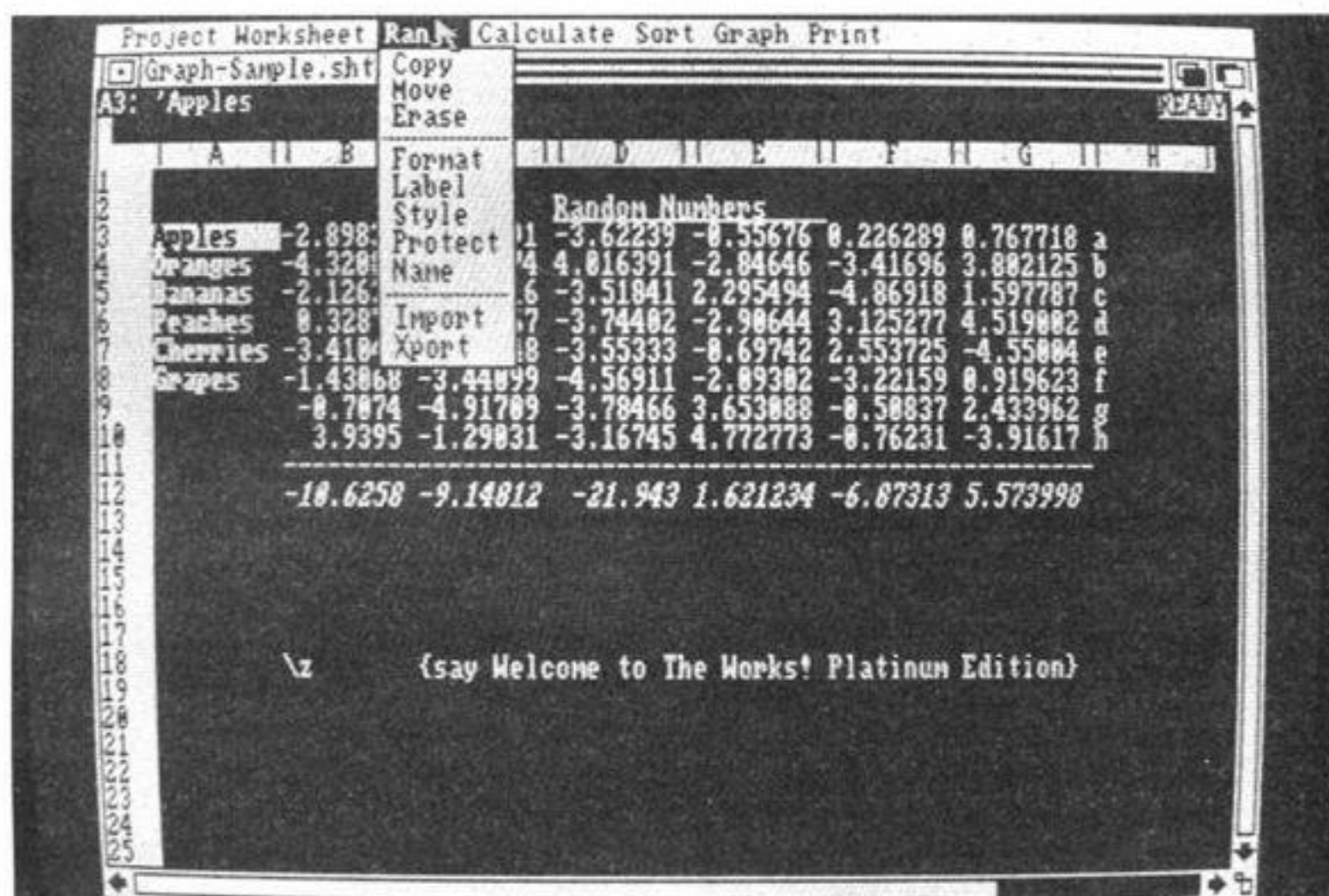
A corredo della PC-502 è inoltre fornito, oltre ad un prontuario di istruzioni sul montaggio (purtroppo in inglese), un dischetto contenente vario materiale di pubblico dominio e, cosa più importante, in una directory di nome **MemTools** (tutta gestibile da workbench), alcuni files che consentono di escludere la memoria aggiuntiva.

Come per tutte le espansioni presenti sul mercato, possono infatti verificarsi alcune incompatibilità (per fortuna sempre più rare) con programmi scritti... diciamo superficialmente. In questi casi, basta un doppio click su **KillFastMem**, ed il computer si resetterà, ripresentadosi con la configurazione di memoria originale.

Anche successivi reset non riattiveranno la "bella dormiente" da 1.5 Mega, fino a che non si spegnerà il computer. Volendo, si puo far resuscitare l'espansione anche via software adoperando un altro file, **GetFastBack**, presente nella stessa directory, che produce l'effetto opposto a quanto prima visto: dopo un reset automatico, il sistema tornerà a "vedere" la ram aggiuntiva.

Davvero un bel regalo per il nostro "ex" 500.

THE WORKS!

a cura di Goran Fisher

Nel numero 76 di C.C.C. abbiamo visto il funzionamento dei moduli di videoscrittura e di telecomunicazione del pacchetto The Works! (platinum edition) della Micro Systems Software (12798 W. Forest Hill Boulevard, Suite 202, West Palm Beach, Florida, USA). Questa volta presentiamo dettagliatamente il funzionamento degli altri due moduli, ovvero del

LO SPREADSHEET

Storicamente, il primo programma di elaborazione elettronica tabellare di grande successo fu **Visicalc** per Apple II. Si può dire che vi fu una sinergia tra i due prodotti, personal computer e software di calcolo, che portò ad una enorme diffusione della microinformatica ad uso personale e, ancora oggi, moltissimi personal computer sono venduti ovunque proprio per essere utilizzati con programmi di questo genere. Il modulo Spreadsheet presente in The Works! consente di strutturare ed elaborare con grande efficacia dati numerici visti come "fogli di lavoro" (**worksheet**) dal programma e di generare automaticamente, per tali gruppi di dati, rappresentazioni grafiche in varie forme.

Menu Project

Il menu Project controlla il caricamento, il salvataggio e la cancellazione dei worksheet, ovvero delle tabelle di dati numerici da manipolare. Vediamo il funzionamento di ogni opzione.

New. Apre un nuovo foglio di lavoro completamente vuoto.

Open. Consente di caricare da un device un file precedentemente salvato con Save.

Appare un requester di file molto semplice da usare, con una lista dei files ordinata e scorrevole.

I gadget sottostanti consentono di cambiare il drive (device) di lettura dei dati e di uscire dal requester. Per caricare un file, basta cliccare due volte sul suo nome, oppure digitarlo nell'apposito requester di stringa sottostante.

Close. Chiude il foglio su cui si sta lavorando. Se il foglio "attuale" non è stato salvato, viene chiesta conferma da un apposito requester che consente di annullare l'operazione.

Save / Save As. Queste funzioni consentono di registrare permanentemente su di un device esterno (floppy disk, hard disk) il foglio sul quale si sta lavorando. Si opera tramite un requester di file praticamente identico a quello di Open, tranne che qui si salva il file invece di caricarlo.

La differenza tra le due funzioni è che la seconda richiede **sempre** di specificare il nome da assegnare al file in un apposito gadget, mentre Save usa l'ultimo nome specificato per il file, od in difetto fa apparire il requester come **Save As**.

Info. Mostra la finestra di regolazione del programma, che riporta tutti i default di funzionamento attuali del programma, come il **metodo** e l'ordine di **ricalcolo**, la **larghezza** di default delle colonne, la prima cella con riferimento circolare ed il **prefisso** delle etichette.

Works. Mostra delle informazioni sulla versione del programma.

Quit. Termina l'esecuzione e rientra al modulo di esecuzione di The Works!

Se il foglio attuale è stato modificato dal momento dell'ultimo salvataggio, viene chiesta conferma.

Menu Worksheet

Le funzioni di questo menu controllano le regolazioni globali di funzionamento del programma.

Insert. Consente di inserire una colonna od una riga nel programma. In questo modo, quando si inserisce una nuova riga / colonna, quelle adiacenti vengono spostate automaticamente. Le formule di calcolo vengono modificate per tenere conto dello spostamento dei dati. Per inserire un gruppo di colonne / righe si deve premere il tasto Return. Se il puntatore di cella non è attualmente collocato sulla colonna / riga da spostare, si può specificare l'indirizzo della colonna (e premere return), oppure scegliere una cella della colonna / riga con il mouse.

Delete. Cancella colonne / righe. In pratica è il complementare della funzione **Insert**, eliminando il gruppo di elementi selezionato, spostando quelli adiacenti per ricoprire lo spazio creato ed aggiustando i riferimenti interni.

Worksheet Column. Fissa il numero di caratteri da inserire in una singola colonna. Il valore di default è pari a **nove** caratteri per colonna, ma tramite questa funzione lo si può fissare tra **1** e **75**. La sottofunzione **Global** consente di modificare la larghezza di default di tutte le colonne nel foglio, che è pari alla larghezza della cella, se non si cambia esplicitamente una colonna o non si riutilizza questo comando. La sottofunzione **Reset** ripristina la colonna sulla quale è collocato il puntatore di posizione alla larghezza fissata come globale.

Worksheet Titles. Un foglio è strutturato con righe e colonne terminanti, solitamente, con almeno una riga (o colonna) di titoli. Ad esempio, la prima riga potrebbe essere una serie di mesi (Gen, Feb, Mar...).

La prima colonna è quasi sempre costituita da celle contenenti etichette di riferimento per l'intera riga cui si riferiscono. Se si fa scorrere lo schermo sopra la prima colonna mostrata potrebbe allora scomparire la riga con le intestazioni dei mesi. La soluzione è bloccare la riga superiore e la prima colonna nello scorrimento, in modo che anche durante il movimento nel foglio oltre i limiti fisici dello schermo, l'intestazione rimanga visualizzata.

Vi sono tre sottofunzioni **Clear**, **Horizontal** e **Vertical**. La prima annulla qualunque bloccaggio di titolo, la seconda blocca tutte le righe sotto quella dove è collocata attualmente il cursore, la terza blocca tutte le colonne a sinistra del cursore.

Format. Qui sono contenute sottofunzioni che consentono di regolare le celle del foglio, con l'eccezione di quelle di intestazione che sono controllate dal comando **Worksheet Label**. Quando si sceglie questa opzione vengono reimpostati tutti i formati indicati con il comando **Range Format** ed il nuovo formato scelto viene applicato immediatamente all'intero foglio . Il formato **General** è quello usato per default; a meno che non lo si modifichi o non lo si riscriva, viene quindi usato per tutto il foglio. Indica numeri interi con tante cifre visualizzate quanto ne consente la larghezza delle colonne. Gli zeri di coda vengono soppressi. Con **Fixed** è consentito specificare il numero di cifre da visualizzare, compreso tra zero e 14 (**2** è il default); ovviamente il numero di cifre effettivamente mostrato dipende anche dalla larghezza delle colonne.

Il formato **Scientific** mostra i numeri in notazione esponenziale, ovvero con una mantissa seguita da una "E" e da un esponente. **Currency** visualizza cifre monetarie secondo la convenzione americana. Il formato "**,**" inserisce virgole nelle cifre scritte, allo scopo di separare le migliaia; se il numero è troppo grande per essere visualizato vengono scritti degli asterischi. Il formato **Percent** mostra valori in percentuale con un numero fisso di cifre decimali, seguiti da un simbolo di percentuale (0.1234 diventa 12.34%). Il formato "**+/-**" crea un grafico a barre positivo o negativo per indicare numeri, solitamente dopo avere duplicato i valori in due colonne per leggere le cifre numeriche accanto alla barra. Il formato **Text** mostra le formule eventualmente presenti nel foglio. Con il formato **Date** è possibile indicare una qualsiasi data del calendario.

Questa viene, in modo "trasparente", automaticamente convertita in un valore numerico (detto numero seriale), per consentire operazioni tra date diverse (esempio: sottrazione tra le due date 12 nov. 1990 - 13 mag. 1988).

Label. Il programma consente di organizzare le etichette di testo (intestazioni) in tre modi entro una data serie di celle. **Left, Center** e **Right**. A ciascun modo corrisponde una sottofunzione: il primo modo allinea a sinistra, il secondo centra, il terzo allinea a destra.

Protect. Consente di proteggere una cella contro le modifiche. Per default, le celle del foglio sono protette, ma il commutatore master è disattivo, in quanto lo stato della protezione viene verificato con il comando **Status**, che mostra i

valori di default per varie regolazioni globali. Per attivare la protezione (**On**) si deve usare la sottofunzione **Protect / Enable**. Ogni cella viene protetta ed è possibile rimuovere la protezione su gruppi individuali di celle. Per disattivare la protezione si usa **Protect / Disable**.

Menu Range

Le funzioni di questo menu consentono di eseguire alcune operazioni su gruppi di celle o sull'intero foglio, a seconda della definizione data per il range.

Copy. Copia il gruppo definito. Le formule vengono appropriatamente ricopiate da una locazione all'altra e corrette affinchè funzionino in base alla nuova posizione degli elementi. Per copiare una cella, si collochi il puntatore sulla cella sorgente, si scelga la funzione del menu e si clicki sulla cella di destinazione.

Move. Funziona come il comando Copy, tranne che il gruppo sorgente viene cancellato durante l'operazione.

Format. Regola le sembianze di una particolare cella, o gruppo di celle, controllate dal **Range Label**. L'operazione di tutte le sottofunzioni è pressochè simile: si sceglie il tipo di formato desiderato, poi il programma richiede il gruppo di celle interessate; l'indicazione può avvenire manualmente o tramite il mouse.

La funzione **Reset** ripristina tutte le regolazioni ai valori di default, ovvero secondo quanto stabilito con Worksheet / Format.

Quanto detto in precedenza, pertanto, vale per i formati **General**, **Fixed**, **Scientific**, **Currency**, **Percent**, **+/-**, **Date**.

Label. Consente di controllare la visualizzazione delle etichette, che possono essere scritte allineate automaticamente a sinistra (funzione **Left**), a destra (**Right**), oppure centrate (**Center**).

Style. Consente di cambiare lo stile di scrittura di un gruppo, ovvero usare il corsivo (**italic**), il grassetto (**bold**) od il sottolineato (**underlined**) per i testi. Con la sottofunzione **Plain** si riporta lo stile nel gruppo indicato a quello "normale".

Protect. Consente di proteggere una gamma di celle contro le modifiche o le cancellazioni accidentali. Se **Global Protection** è **Off**, le celle non possono essere protette, mentre se è **On** è consentito protegge o sproteggere dei gruppi individuali tramite questo comando.

Name. Consente di denominare una gamma di celle per un riferimento più rapido. La funzione **Create** consente di definire il nome, mentre le sottofunzioni **Left, Right, Below** e **Above** consentono, rispettivamente, di indicare che le celle da denominare sono a sinistra, a destra, sotto o sopra quella attualmente indicata dal cursore. La sottofunzione **Delete** consente di cancellare il nome di un gruppo tramite apposito requester, senza toccare il contenuto delle celle. Compito, questo, svolto invece da **Erase**, che cancella il contenuto di tutte le celle del gruppo indicato per nome.

Menu Calculate

Il menu Calculate contiene comandi che consentono di eseguire calcoli sugli elementi del foglio. Per forzare il calcolo del foglio attuale in qualunque momento si può premere il tasto **F9**.

Method. I sottocomandi di questo menu consentono di modificare il modo di ricalcolo dello Spreadsheet. Con **Manual** lo spreadsheet esegue il ricalcolo solo quando lo si richiede espressamente con **F9**.

Con **Auto**, invcece, il ricalcolo viene eseguito tutte le volte che si modifica o si immette un valore.

Order. I sottocomandi di questo menu fissano l'ordine con il quale il programma esegue il ricalcolo. Con **Natural** la ricerca dell'ordine delle espressioni da valutare procede in modo da produrre un risultato per ogni cella. Con **Rowwise** l'ordine è rigorosamente dall'alto verso il basso delle righe, mentre con **Columnwise** procede da sinistra verso destra, per colonne. Con **Iteration** è possibile indicare quante volte (tra 1 e 50) va ricalcolato il foglio, indispensabile nel caso di fogli automodificanti (una cella può essere generatrice di un seme casuale per altre celle). Con **Alert** il programma genera un segnale acustico ogni volta che termina il calcolo.

Menu Sort

Le funzioni del menu Sort sono utilizzabili per riordinare un gruppo di dati secondo un criterio **ascendente** o **discendente**. Sono accettabili come dati anche celle vuote, lettere, formule o numeri.

Sort Data Range. Consente di definire il gruppo di dati da riordinare.

Sort Primary Key. Consente di indicare la prima colonna di riordine. Dopo averla scelta, compare un requester che

richiede il tipo di riordino (Ascendente o Discendente).

Sort Secondary Key. Consente di indicare la colonna secondaria di riordino, ovvero quella che determina come vengono riordinate sequenze in cui le prime colonne sono identiche.

Sort Go. Avvia il riordino.

Sort Reset. Azzera tutte le regolazioni delle funzioni del menu Sort.

Menu Graph

Questo menu contiene i comandi necessari per definire la memorizzazione e la visualizzazione dei grafici. Si possono visualizzare sino a quattro grafici per volta.

Graph Load. L'opzione di caricamento grafico carica il requester di File che visualizza i files "**.gdf**". Questi sono i files di definizione del grafico, che consente di memorizzare sino a quattro grafici per foglio.

Graph Save. Registra su memoria di massa (disco) i grafici e le loro definizioni.

Graph View. Traccia il grafico predefinito (od i grafici, se sono più di uno).

Graph Number. Fissa e visualizza le regolazioni per un massimo di quattro grafici. Il numero di grafico spuntato rifletterà le regolazioni grafiche che sono state visualizzate nel menu Graph. Le sottofunzioni del menu definiscono le opzioni per vedere il primo, secondo, terzo e quarto grafico.

Model. Determina come verranno visualizzati i dati. Nel caso non si avesse molta familiarità con questo tipo di rappresentazioni, si provi a visionarle tutte. Il modo **Pie** genera un cosiddetto grafico "a torta", con al massimo 10-15 elementi. Il modo **Bar** usa un grafico a barre per visualizzare i dati. Il modo **Line** usa una linea non spezzata per ciascun gruppo di elementi. Il modo **X-Y** usa due gruppi di dati e deve quindi avere almeno due gruppi di punti per apparire effettivamente. Il modo **Stk-Bar** usa un grafico a barre sovrapposte. Il modo **Z-Pie** usa una specie di grafico a torta, ma tridimensionale. Infine il modo **3D-Bar** usa barre tridimensioanli ed è l'unico modo in cui l'elemento grafico rappresentante un gruppo di dati può essere coperto da un altro.

Labels. Consente di intestare i dati rappresentati, secondo tre tipi di etichette ("labels"). Con **Legend** si identificano i ranges nel grafico, con **Group** si identificano i groups dei ranges e con le sei sottofunzioni seguenti (**A...F**) si definiscono materiamente le intestazioni.

Titles. Definisce l'intestazione per finestra del grafico. Si possono avere uno o due titoli per ogni grafico e uno per ogni asse (lungo la linea inferiore o sul lato).

Con la sottofunzione **Clear** si cancellano tutte le intestazioni definite con questo menu. Con **First** e **Second** si localizzano le posizioni del primo e del secondo titolo sulla finestra, rispettivamente. Con **X-Axis**, **Y-Axis** e **Z-Axis** il titolo apparirà, rispettivamente, lungo la linea inferiore del grafico, lungo il bordo sinistro e (solo per il modo 3D-Bar) lungo il bordo destro.

Scale. Consente di scalare normalmente il grafico in modo automatico o manuale; in quest'ultimo caso, bisogna specificare il punto inferiore e superiore del sistema di riferimento del grafico (**Automatic** o **Manual**). Con **Low- Limit** e **High-Limit** si specificano, appunto, il punto di partenza ed il punto di fine del grafico quando si usa la scalatura manuale.

Options. Le opzioni di questo menu contengono una collezione di funzioni che influenzano l'aspetto del grafico. Con **Clear** si azzerano le regolazioni effettuate, **Lines** forza la tracciatura di linee continue che connettono i punti dei grafici in modo "Line" e "X-Y", **Symbols** traccia questi due tipi di grafici con il simbolo localizzato alla coordinata specificata dal ranges selezionato, **X-Grid** e **Y-Grid** tracciano automaticamente delle griglie per gli assi X e Y e **Z-Grid** attiva la griglia per l'asse Z nel grafico di tipo "3D Bar".

Clear. Annulla alcune o tutte le regolazioni fatte per un grafico. Il comando **All** cancella tutte le definizioni, **Data-Ranges** ripristina tutti i ranges di dati appena selezionati, **Element-Labels** cancella tutte le etichette, **Legend-Labels** cancella le etichette di lettura e **Group Labels** le etichette dei gruppi.

Menu Print

I comandi di questo menu consentono di stampare intestazioni, numeri di pagine e headers e footers nei documenti ("reports") prodotti dal programma.

Range. Definisce il range di stampa nella stampa di un rapporto.

Go. I sottocomandi qui presenti iniziano la stampa. Con **Printer** l'output del programma viene inviato alla stampante in base alle regolazioni fatte dal menu Print, mentre con **File** si invia l'output ad un file di cui si indica il nome.

Margins. Qui sono presenti i sottocomandi che consentono di specificare la lunghezza della pagina (**Page Length**), la linea superiore ed inferiore (**Top e Bottom**) ed i margini sinistro e destro (**Left e Right**) della stampa.

Borders. Si possono indicare delle

etichette di righe (**Rows**) o colonne (**Columns**) come bordi per le stampe.

Options. Regola il modo di stampa di headers, footers e la stringa di inizializzazione (**setup**) della stampante. Con **Clear** si azzerano tutte le regolazioni fatte in questo menu. Con **As-Displayed** le celle del foglio vengono stampate esattamente come appaiono sullo schermo, inclusi eventuali stili speciali. Con **Use-Margins** vengono usati tutti i valori specificati con la funzione **Print / Margins**.

Con i comandi **Header** e **Footer** vengono prodotte automaticamente delle stringhe che iniziano e terminano la stampa per ogni foglio. Con **Setup** si può immettere la stringa di inizializzazione che viene inviata alla stampante prima dei dati veri e propri.

Graph. Stampa una hard-copy di quanto è visibile attualmente sullo schermo.

Reset. Azzera completamente od in parte le opzioni di stampa. Con **All** si annullano tutte le regolazioni, con **Borders** si cancellano solo le regolazioni dei bordi, con **Margins** quelle dei margini, con **Range** si annulla il range di stampa specificato.

IL DATABASE

Gli archiviatori sono un'altra applicazione principe dei computer, indispensabili per gestire velocemente archivi senza ricorrere alla consultazione di libroni ed agende. Il pacchetto The Works! comprende un database di ottima fattura, molto semplice da usare, ma dotato di tutte le funzioni e le caratteristiche necessarie per essere utile a chiunque abbia voglia di gestire anche qualcosa di molto più complesso di una semplice lista di indirizzi di amici o di programmi per Amiga.

Menu Project

Le funzioni di questo menu sono preposte essenzialmente al controllo del lavoro globale e dei files dell'archivio.

New. Crea un nuovo database azzerando completamente la memoria di lavoro.

Open. Fa comparire un requester di file che consente di **caricare** un file di archiviazione. La lista è scorrevole e l'indicazione può essere data cliccando due volte col mouse sul nome interessato oppure digitando da tastiera il nome stesso nel gadget di stringa e cliccando poi sul gadget Open (oppure premento return a completamento del nome e poi "**O**").

Close. Chiude tutti i files aperti ed il database corrente.

Save / Save As. Salvano l'archivio corrente sulla memoria di massa (disco). Save As fa comparire sempre il requester, simile a quello di Open, che consente di digitare il nome con il quale verrà salvato il file. Save, invece, salva direttamente usando il nome precedente, se specificato, oppure fa comparire anche esso il requester se non c'è alcun nome definito.

Change. Modifica un database esistente.

Lo si usi con prudenza, perchè alcuni dati **possono andare persi** quando si cambiano i tipi di campi del database, si rimuovono campi o si decrementano le loro larghezze.

Copy. Crea un nuovo database prendendo le stesse definizioni usate per uno già esistente, ricopiandolo con un nuovo nome e svuotandolo dei contenuti (**Design**), oppure ricopiando i record che soddisfano ad un certo "filtro" (**Selected**), ma comunque ignorando gli indici ed i formati definiti.

Info. Produce informazioni sul datatabse attuale.

Works. Produce informazioni sul programma.

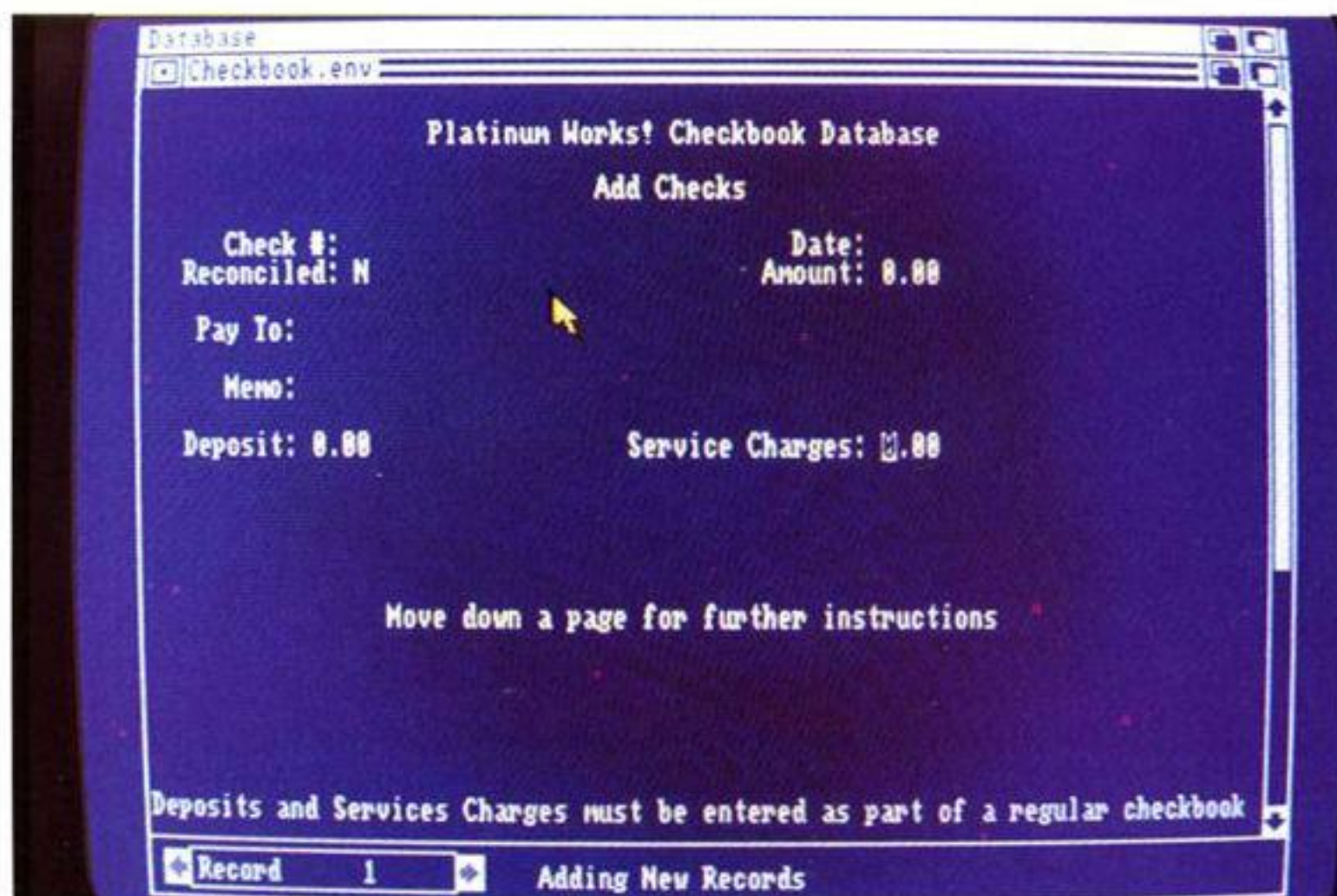
Quit. Termina l'esecuzione del modulo database e rientra al gestore principale. Chiede conferma prima di eseguire l'operazione nel caso che il database attuale sia stato modificato dal momento dell'ultimo salvataggio, consentendo di salvarlo tramite l'apposito requester di file.

Menu Index

Gli indici sono usati per consentire la ricerca rapida ed il riordino dei records. Contiene tutte le funzioni necessarie per aprire, chiudere, creare e cancellare indici. Si può aprire un solo file indice per volta.

New. Crea un nuovo Index per un certo denominato campo. Se è già stato aperto un indice, non ne può essere creato un altro.

L'indice deve essere creato prima dell'uso di ogni Report che richiede riordino dei campi.

Open. Fa comparire un requester con l'elenco degli indici esistenti. Consente quindi di accedere ad un indice esistente. Questa opzione non è attivabile (**gho-**

sted) se è già aperto un indice.

Close. Chiude l'indice corrente.

Menu Form

Qui sono contenute le funzioni necessarie al mantenimento del database.

Load. Mostra un requester per il caricamento di formati personalizzati per l'immissione e la rilettura dei dati dell'archivio.

Save. Consente di salvare, tramite requester, un formato personalizzato.

Change. Porta ad un altro schermo dove si possono creare maschere di immissione e rilettura dei dati personalizzate.

L'uso delle funzioni riportate in questo schermo è molto intuitivo.

Menu Search

Questo menu contiene le opzioni necessarie per reperire dei record in un database.

Find Next / Prev. Ricercano la ricorrenza successiva o precedente, rispettivamente, del record che concorda col filtro di ricerca attuale.

Browse. Questa opzione è disponibile solo se è aperto un indice sul file. Ricerca l'indice corrente per le esatte concordanze tra quanto si cerca e quanto vi è nell'archivio, ignorando le dimensioni dei caratteri.

Define Filter (1 - 4). Consente di fissare sino a 4 criteri di ricerca dei dati. Si può comunque usare un solo filtro per volta e può essere usato sia per le ricerche regolari nel database, sia per la generazione di rapporti.

Search Use Filter. Sono usati per scegliere quale filtro usare. Sono accessibili nel menu solo quelli definiti.

Menu Transfer

Le funzioni di questo menu usano il **Clipboard.Device** di Amiga per importare ed esportare files Mailmerge, che vengono "stampati" in un file su disco oppure sul Clipboard.

Import. Legge un file in formato Mailmerge, ovvero un file ASCII con i campi in ordine sequenziale separati da virgole e con dei CR che terminano ogni record. **File** legge il database da un file, **Clipboard** tramite il Clipboard.Device.

Export. Scrive l'archivio in formato Mailmerge in un file su disco (Export / File) oppure nel Clipboard (Export Clipboard).

Sort. Consente di riordinare in modo ascendente, discendente o casuale (**Ascending, Descending, None**), secondo il file indice, l'archivio.

Filter. Quando è stato definito un filtro di ricerca, è possibile attivare o disattivare (**On / Off**) tale filtro da qui per discriminare i files che verranno effettivamente esportati.

Menu Print

Questo menu contiene, come prevedibile, le funzioni di controllo della stampa di rapporti su carta dell'archivio.

Preview. Presenta sullo schermo il report.

Go. Produce verso un file o verso la printer il report.

Forms. Stampa usando il corrente Formato.

Design. Stampa la sagoma del Formato attuale, insieme alle eventuali formule, più un listato dei nomi di campi e delle corrispondenti etichette.

Options. Tramite le sue sottofunzioni, fissa la **lunghezza** del foglio di stampa, il **margine** superiore ed inferiore, la stringa di **setup** inviata alla stampante prima dei dati, il tipo di **blanks** da usare (-1 invia un Form-Feed tra i formati, 0 non lascia alcuno spazio).

Sort. Quando è aperto un indice, consente di stabilire l'ordine di stampa dei dati tra ascendente, discendente e casuale.

Filter. Attiva o disattiva il filtro di ricerca, se definito, che discrimina i record effettivamente inviati alla stampa.

Sul n. **76** di C.C.C. è stata pubblicata la prima parte della presentazione di "**The Works!**". Ricordiamo ai lettori interessati che il pacchetto è uno dei più potenti e versatili oggi disponibili per Amiga dal momento che comprende, oltre allo **Spreadsheet** ed al **Database** di cui si parla in queste pagine, un **Word Processor** ed un programma di **Telecomunicazione** utile per effettuare scambi di dati servendosi del Modem e, quindi, della normale linea telefonica.

QUANTO COSTA IL TUO COMMODORE

Amiga 2000 - L. 2.715.000

Microprocessore Motorola MC68000 - Clock 7.16MHz - Kickstart ROM - Memoria RAM: 1 MByte - 3 chip custom per DMA, Video, Audio, I/O - 5 Slot di Espansione Amiga Bus 100 pin Autoconfig™ - 1 Slot di Espansione 86 pin per Schede Coprocessore - 2 Slot di Espansione compatibili AT/XT - 2 Slot di Espansione compatibili XT - 2 Slot di Espansione Video - 1 Floppy Disk Drive da 3 1/2", 880 KBytes - Porta seriale RS232C - Sistema Operativo single-user, multitasking AmigaDOS - Compatibilità MS-DOS XT/AT disponibile con schede interne Janus (A2088 - A2286) - Monitor escluso

Amiga 500 - L. 995.000

Microprocessore Motorola MC68000 - Clock 7.16 MHz - Kickstart ROM - Memoria RAM: 512 KBytes - 3 Chip custom per DMA, Video, Audio, I/O - 1 Floppy Disk Driver da 3 1/2", 880 KBytes - Porta seriale RS232C - Porta parallela Centronics

Videomaster 2995 - L. 1.200.000

Desk Top Video - Sistema per elaborazini video semiprofessionale composto da genlock, digitalizzatore e alloggiamento per 3 drive A2010 - Ingressi videocomposito (2), RGB - Uscite Videocomposito, RF, RGB + sync -

Floppy Disk Driver A 1010 - L. 335.000

Floppy Disk Driver - Drive esterno da 3 1/2" - Capacità 880 KBytes - Collegabile a tutti i modelli della linea Amiga, alla scheda A2088 e al PC1

Floppy Disk Drive A 2010 - L. 280.000

Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" - Capacità 880 KBytes - Collegabile ad Amiga 2000

Hard Disk A 590 - L. 1.750.000

Hard Disk+Controller+RAM - Scheda Controller - Hard Disk da 3 1/2" 20 MBytes - 2 MBytes "fast" RAM - Collegabile all'Amiga 500

Scheda Janus A 2088 + A 2020 - L. 1.050.000

Scheda Janus XT+Floppy Disk Drive da 5 1/4", 360 KBytes - Scheda Bridgeboard per compatibilità MS-DOS (XT) in Amiga 2000 - Microprocessore Intel 8088 - Coprocessore matematico opzionale Intel 8087

A2286+A2020 - L. 1.985.000

Scheda Janus AT+Floppy Disk Drive da 5 1/4", 1.2 MBytes - Scheda Bridgeboard per compatibilità MS-DOS (AT) in Amiga 2000 - Microprocessore Intel 80287 - Clock 8 MHz - RAM: 1 MBytes on-board - Floppy Disk Controller on-board - Floppy Disk Driver disegnato per l'installazione all'interno dell'Amiga 2000 -

Scheda A2620 - L. 2.700.000

Scheda Processore Alternativo 32 bit - Scheda per 68020 e Unix - Microprocessore Motorola MC68020 - Coprocessore matematico Motorola MC68881 (opzionale MC68882)

Scheda A Unix - L. 3.250.000

Sistema Operativo AT&T Unix System V Release 3 - Per Amiga 2000 con scheda A2620 e Hard Disk 100 MBytes

Hard Disk A2092+PC5060 - L. 1.020.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+2092 - L. 1.240.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+A2094 - L. 1.900.000

Stesse caratteristiche del kit A2092 ma con disco da 40 MBytes

Espansione di memoria A2058 - L. 1.149.000

Espansione di memoria - Scheda di espansione per Amiga 2000 - Fornita con 2 MBytes "fast" RAM, espandibile a 4 o 8 MBytes

Scheda Video A2060 - L. 165.000

Modulatore video - Scheda modulatore video interna per Amiga 2000 - Uscite colore e monocromatica - Si inserisce nello slot video dell'Amiga 2000

Genlock Card A2301 - L. 420.000

Genlock - Scheda Genlock semiprofessionale per Amiga 2000 - Permette di miscelare immagini provenienti da una sorgente esterna con immagini provenienti dal computer

Professional Video Adapter Card A2351 - L. 1.500.000

Professional Video Adapter - Scheda Video Professionale per Amiga 2000 (B) - Genlock qualità Broadcast - Frame Grabber - Digitalizzatore - Include software di controllo per la gestione interattiva
(Disponibile da maggio '89)

A501 - L. 300.000

Espansione di memoria - Cartuccia di espansione di memoria da 512 KBytes per A500

A520 - L. 45.000

Modulatore RF - Modulatore esterno A500 - Permette di connettere qualsiasi televisore B/N o colori ad Amiga 500

A Scart - L. 27.000

Cavo di collegamento A500/A2000 con connettore per televisione SCART

Monitor a colori 1084 - L. 595.000

Monitor a colori ad alta risoluzione - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor a colori 2080 - L. 770.000

Monitor a colori ad alta risoluzione e lunga persistenza - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Frequenza di raster 50 Hz - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor Monocromatico A2024 - L. 1.235.000

Monitor monocromatico a fosfori "bianco-carta" - Turbo 14" antiriflesso - (Disponibile da marzo '89)

PC60/40 - L. 7.812.000

Microprocessore Intel 80386 - Coprocessore matematico opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 funzioni - Sistema Operativo MS-DOS 3 2.1 - Interprete GW-Basic

PC60/40C - L. 8.127.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 60/80 - L. 10.450.000

Microprocessore Intel 80386 - Coprocessore opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Memoria RAM: 2.5 MBytes - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Floppy Disk Drive opzionale da 3 1/2", 1.44 MBytes - 1 Hard Disk da 80 MBytes - 2 Porte parallele Centronics - Mouse video EGA (compatibile MDA - Hercules - CGA). Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Ambiente Operativo Microsoft Windows/386 - Interprete GW-Basic

PC60/80C - L. 10.700.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC40/20 - L. 4.100.000

Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/20C - L. 4.350.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 40/40 - L. 5.285.000

Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/40C - L. 5.535.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

1352 - L. 78.000

Mouse - Collegabile con Microsoft Bus Mouse - Collegabile direttamente a PC1, PC10/20 - III, PC40 - III

PC910 - L. 355.000

Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" per PC10/20-I-II-III - Capacità 360 o 720 KBytes selezionabile tramite "config. sys" - Corredo di telaio di supporto per l'installazione in un alloggiamento per un drive da 5 1/4" - Interfaccia identica ai modelli da 5 1/4"

PC1 - L. 995.000

Microprocessore Intel 8088 - 1 Floppy Disk Drive da 5 1/4" - Porta seriale RS232C - Porta parallela Centronics - - Monitor monocromatico 12" - Tastiera 84 tasti - Sistema Operativo MS-DOS 3.2 - Interprete GW-Basic

PCEXP1 - L. 640.000

PC Expansion Box - Box esterno di espansione per PC 1 - Alimentatore aggiuntivo incluso - Contiene 3 Slot di Espansione compatibili Ibm XT - Alloggiamento per Hard Disk da 5 1/4" - Si posiziona sotto il corpo del PC1 e viene collegato tramite degli appositi connettori

PC10-III - L. 1.360.000

Microprocessore Intel 8088 Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - Memoria RAM: 640 KBytes - 2 Floppy Disk Drive da 5 1/4", 360 KBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC10-IIIC - L. 1.675.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC20-III - L. 2.095.000

Microprocessore Intel 8088 - Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - 1/4", 360 KBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse) - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC20-IIIC - L. 2.410.000
Stessa configurazione ma con monitor 14" a colori mod. 1084

Nuovo C64 - L. 325.000
Nuovo Personal Computer CPU 64 KBytes RAM - Vastissima biblioteca software disponibile - Porta seriale Commodore - Porta registratore per cassette - Porta parallela programmabile -

C128D - L. 895.000
Personal Computer CPU 128 KBytes RAM espandibile a 512 KBytes - ROM 48 KBytes - Basic 7.0 - Tastiera separata - Funzionante in modo 128,64 o CP/M 3.0 - Include floppy disk drive da 340 KBytes

Floppy Disk Drive 1541 II - L. 365.000
Floppy Disk Drive - Floppy Disk Drive da 5 1/4" singola faccia - Capacità 170 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

Floppy Disk Dirve 1581 - L. 420.000
Floppy Disk Drive da 3 1/2" doppia faccia - Capacità 800 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

1530 - L. 55.000
Registratore a cassette per C64, C128, C128D

Accessori per C64 - 128D
1700 - Espansione di memoria - Cartuccia di espansione di memoria a 128 KBytes per C128 **- L. 170.000**

1750 - Espansione di memoria - Cartuccia di espansione di memoria 512 KBytes per C128 - **L. 245.000**

1764 - Espansione di memoria - Cartuccia di espansione di memoria a 256 KBytes per C64 - Fornita di alimentatore surdimensionato - **L. 198.000**

16499 - Adattatore Telematico Omologato - Collegabile al C64 - Permette il collegamento a Videotel, P.G.E. e banche dati **L. 149.000**

1399 - Joystick - Joystick a microswitch con autofire - **L. 29.000**

1351 - Mouse - Mouse per C64, C128, C128D - **L. 72.000**

Monitor Monocromatico 1402 - L. 280.000
Monitor monocromatico a fosfori "bianco-carta" - Turbo 12" antiriflesso - Ingresso TTL - Compatibile con tutta la gamma PC

Monitor Monocromatico 1404 - L. 365.000
Monitor monocromatico a fosfori ambra - Turbo 14" antiriflesso a schermo piatto - Ingresso TTL - Compatibile con tutta la gamma PC - Base orientabile

Monitor Monocromatico 1450 - L. 470.000
Monitor monocromatico BI-SYNC a fosfori "bianco-carta" - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Monitor a colori 1802 - L. 445.000
Monitor a colori - Turbo 14" - Collegabile a C64, C128, C128D

Monitor monocromatico 1900 - L. 199.000
Monitor monocromatico a fosfori verdi - Turbo 12" antiriflesso - Ingresso videocomposito - Compatibile con tutta la gamma Commodore

Monitor a colori 1950 - L. 1.280.000
Monitor a colori BI-SYNC alta risoluzione - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Stampante MPS 1230 - L. 465.000
Stampante a matrice di punti - Testina a 9 aghi - 120 cps - Bidirezionale - 80 colonne - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

MPS 1230R - L. 19.000
Nastro per stampante

Stampante MPS 1500C - L. 495.000
Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia parallela Centronics - Compatibile con la gamma Amiga e PC

MPS1500R - L. 37.000
Nastro a colori per stampante

MPS1500R - L.37.000
Nastro a colori per stampante

Stampante MPS 1550C - L. 575.000
Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

I COMMODORE POINT

LOMBARDIA

Milano
• AL RISPARMIO - V.LE MONZA 204
• BCS - VIA MONTEGANI 11
• BRAHA A. - VIA PIER CAPPONI 5
• E.D.S. - C.SO PORTA TICINESE 4
• FAREF - VIA A. VOLTA 21
• FLOPPERIA - V.LE MONTENERO 31
• GBC - VIA CANTONI 7 - VIA PETRELLA 6
• GIGLIONI - V.LE LUIGI STURZO 45
• L'UFFICIO 2000 - VIA RIPAMONTI 213
• LOGITEK - VIA GOLGI 60
• LU - MEN - VIA SANTA MONICA 3
• MARCUCCI - VIA F.LLI BRONZETTI 37
• MELCHIONI - VIA P. COLLETTA 37
• MESSAGGERIE MUSICALI - GALLERIA DEL CORSO 2
• NEWEL - VIA MAC MAHON 75
• PANCOMMERZ ITALIA - VIA PADOVA 1
• SUPERGAMES - VIA VITRUVIO 38
• 68000 E DINTORNI - VIA WASHINGTON 91
Provincia di Milano
• GINO FERRARI CENTRO HI-FI - VIA MADRE CABRINI 44 - S. ANG. LODIGIANO
• F.LLI GALIMBERTI - VIA NAZIONALE DEI GIOVI 28/36 - BARLASSINA
• TECNOLUX - VIA PIETRO NENNI 5 - BERNATE TICINO
• OGGIONI & C. - VIA DANTE CESANA 27 - CARATE BRIANZA
• AL RISPARMIO - VIA U. GIORDANO 57 - CINISELLO BALSAMO
• GBC - V.LE MATTEOTTI 66 - CINISELLO BALSAMO
• CASA DELLA MUSICA - VIA INDIPENDENZA 21 - COLOGNO MONZESE
• PENATI - VIA VERDI 28/30 - CORBETTA
• EPM SYSTEM - V.LE ITALIA 12 - CORSICO
• P.G. OSTELLARI - VIA MILANO 300 - DESIO
• CENTRO COMPUTER PANDOLFI - VIA CORRIDONI 18 - LEGNANO
• COMPUTEAM - VIA VECELLIO 41 - LISSONE
• M.B.M. - C.SO ROMA 112 - LODI
• L'AMICO DEL COMPUTER - VIA CASTELLINI 27 - MELEGNANO
• BIT 84 - VIA ITALIA 4 - MONZA
• IL CURSORE - VIA CAMPO DEI FIORI 35 - NOVATE MIL.
• I.C.O. - VIA DEI TIGLI 14 - OPERA
• R & C ELGRA - VIA SAN MARTINO 13 - PALAZZOLO MIL.
• ESSEGIEMME SISTEMI SAS - VIA DE AMICIS 24 - RHO
• TECNO - CENTRO - VIA BARACCA 2 - SEREGNO
• NIWA HARD&SOFT - VIA B. BUOZZI 94 - SESTO SAN GIOV.
• COMPUTER SHOP - VIA CONFALONIERI 35 - VILLASANTA
• ACTE - VIA B. CREMIGNANI 13 - VIMERCATE
• IL COMPUTER SERVICE SHOP - VIA PADANA SUPERIORE 197 - VIMODRONE
Bergamo
• D.R.B. - VIA BORGO PALAZZO 65
• TINTORI ENRICO &C. - VIA BROSETA 1
• VIDEO IMMAGINE - VIA CARDUCCI c/o CITTA' DI MERCATO
Provincia di Bergamo
BERTULEZZI GIOVANNI - VIA FANTONI 48 - ALZANO LOMBARDO
• COMPUTER SHOP - VIA VITTORIO VENETO 9 - CAPRIATE SAN GERVASIO
• B M R - VIA BUTTARO 4/T - DALMINE
• MEGABYTE 2 - VIA ROMA 61/A - GRUMELLO
• OTTICO OPTOMETRISTA ROVETTA - P.ZZA GARIBALDI 6 - LOVERE
• COMPUTER POINT - VIA LANTIERI 52 - SARNICO
• A.B. INFORMATICA - STRADA STATALE CREMASCA 66 - URGNANO
Brescia
• MASTER INFORMATICA - VIA F.LLI UGONI 10/B
PROVINCIA DI BRESCIA
• MISTER BIT - VIA MAZZINI 70 - BRENO
• CAVALLI PIETRO - VIA 10 GIORNATE 14 BIS - CASTREZZATO
• VIETTI GIUSEPPE - VIA MILANO 1/B - CHIARI
• MEGABYTE - P.ZZA MALUEZZI 14 - DESENZANO DEL GARDA
• BARESI RINO &C. - VIA XX SETTEMBRE 7 - GHEDI
• INFO CAM - VIA PROVINCIALE 3 - GRATACASOLO
• "PAC-LAND" di GARDONI - CENTRO COM.le - LA CASA DI MARGHERITA D'ESTE - VIA GIORGIONI 21
Como
• IL COMPUTER - VIA INDIPENDENZA 90
• 2M ELETTRONICA - VIA SACCO 3
Provincia di Como
• ELTRON - VIA IV NOVEMBRE 1 - BARZANO'
• DATA FOUND - VIA A. VOLTA 4 -ERBA
• CIMA ELETTRONICA - VIA L. DA VINCI 7 - LECCO
• FUMAGALLI - VIA CAIROLI 48 -LECCO
• RIGHI ELETTRONICA - VIA G. LEOPARDI 26 - OLGIATE COMASCO
Cremona
• MONDO COMPUTER - VIA GIUSEPPINA 11/B
• PRISMA - VIA BUOSO DA DOVARA 8
• TELCO - P.ZZA MARCONI 2/A
Provincia di Cremona
• ELCOM - VIA IV NOVEMBRE 56/58 - CREMA
• EUROELETTRONICA - VIA XX SETTEMBRE 92/A - CREMA
Mantova
• COMPUTER CANOSSA - GAL. FERRI 7
• 32 BIT - VIA C. BATTISTI 14
• ELET. di BASSO - V.LE RISORGIMENTO 69
Provincia di Mantova
• CLICK - ON COMPUTER - S.S. GOITESE 168 - GOITO
Pavia
• POLIWARE - C.SO C. ALBERTO 76
• SENNA GIANFRANCO - VIA CALCHI 5
Provincia di Pavia
• A. FERRARI - C.SO CAVOUR 57 - MORTARA
• LOGICA MAINT - V.LE M.TE GRAPPA 32 - VIGEVANO
• M. VISENTIN - C.SO V. EMANUELE 76 - VIGEVANO
Sondrio
• CIPOLLA MAURO - VIA TREMOGGE 25
Provincia di Sondrio
• FOTONOVA - VIA VALERIANA 1 - S.PIETRO DI BERBENNO
Varese
• ELLE - EFFE - VIA GOLDONI 35
• IL C.TRO ELET. - VIA MORAZZONE 2
• SUPERGAMES - VIA CARROBBIO 13
Provincia di Varese
• BUSTO BIT - VIA GAVINANA 17-BUSTO A.
• MASTER PIX - VIA S.MICHELE 3 - BUSTO A.
• PUNTO UFFICIO - VIA R.SANZIO 8 - GALLARATE
• GRANDI MAGAZZINI BOSSI - VIA CLERICI 196 - GERENZANO
• J.A.C. - C.so MATTEOTTI 38 - SESTO C.

PIEMONTE
Alessandria
• BIT MICRO - VIA MAZZINI 102
• SERV. INFOR. - VIA ALESSANDRO III 47
Provincia di Alessandria
• SONY ITALIANA - VIA G. MANARA 7 - CASALE MONFERRATO
• SGE ELETTRONICA - VIA BANDELLO 19 - TORTONA
• COMPUTER TEMPLE - VIA F. CAVALLOTTI 13 - VALENZA
Asti
• ASTI GAMES - C.SO ALFIERI 26
• RECORD - C.SO ALFIERI 166/3 (Galleria Argenta)
Cuneo
• ROSSI COMPUTERS - C.SO NIZZA 42
Provincia di Cuneo
• PUNTO BIT - C.SO LANGHE 26/C - ALBA
• BOSETTI - VIA ROMA 149 - FOSSANO
• COMPUTERLAND - VIA MAZZINI 30/32 - SALUZZO
Novara
• PROGRAMMA 3 - V.LE BUONARROTI 8
• PUNTO VIDEO - C.so RISORGIMENTO 39/B
Provincia di Novara
• COMPUTER - VIA MONTE ZEDA 4 -ARONA
• ALL COMPUTER - C.SO GARIBALDI 106 - BORGOMANERO
• S.P.A. - C.SO DISSEGNA 21/BIS - DOMODOSSOLA
• ELLIOTT COMPUTER SHOP - VIA DON MINZONI 32 - INTRA
• TRISCONI VALERIA - VIA MAZZINI 90 - OMEGNA
Torino
• ABA ELETTRONICA - VIA C. FOSSATI 5/P
• ALEX COMPUTER E GIOCHI - C.SO FRANCIA 333/4
• COMPUTER HOME - VIA SAN DONATO 46/D
• COMPUTING NEW - VIA M. POLO 40/E
• C.D.M. ELETTR. - VIA MAROCHETTI 17
• DE BUG - C.SO V. EMANUELE II 22
• DESME UNIVERSAL - VIA S.SECONDO 95
• FDS ALTERIO - VIA BORGARO 86/D
• IL COMPUTER - VIA N. FABRIZI 126
• MICRONTEL - C.SO D. degli ABRUZZI 28
• PLAY GAMES SHOP - VIA C. ALBERTO 39/E
• RADIO TV MIRAFIORI - C.SO UNIONE SOVIETICA 381
• SMT ELETTRONICA - VIA BIBIANA 83/bis
Provincia di Torino
• PAUL E CHICO VIDEOSOUND - VIA V.EMANUELE 52 - CHIERI
• BIT INFORMATICA - VIA V. EMANUELE 154 - CIRIE'
• HI - FI CLUB - C.SO FRANCIA 92C - COLLEGNO
• MISTER PERSONAL - VIA CATTANEO 52 - FAVRIA
• I.C.S. - VIA TORINO 73 - IVREA
• DAG - VIA I MAGGIO 40 - LUSERNA S. GIOVANNI
• EUREX - C.SO INDIPENDENZA 5 - RIVAROLO CANAVESE
• DIAM INFORMATICA - C.SO FRANCIA 146 bis - RIVOLI
• FULLINFORMATICA - VIA V.VENETO 25 - RIVOLI
• GAMMA COMPUTER - VIA CAVOUR 3A-3B - SET.TORINESE
Vercelli
• ELETTROGAMMA - C.SO BORMIDA 27 ang. V.Montanara
• ELETTRONICA - STRADA TORINO 15
Provincia di Vercelli
• C.S.I. TEOREMA - VIA LOSANA 9 - BIELLA
• SIGEST - VIA BERTODANO 8 - BIELLA
• REMONDINO FRANCO - VIA ROMA 5 - BORGOSESIA
• FOTOSTUDIO TREVISAN - VIA XXV APRILE 24/B - COSSATO
• STUDIO FOTOGRAFICO IMARISIO - P.ZZA M. LIBERTA' 7 - TRINO

VENETO
Belluno
• UP TO DATE - VIA V. VENETO 43
Provincia di Belluno
• GUERRA COMPUTERS - V.LE MAZZINI 10/A - FELTRE
Padova
• BIT SHOP - VIA CAIROLI 11
• COMPUMANIA - VIA T. CAMPOSANPIERO 37
• D.P.R. DE PRATO R. - V.LO LOMBARDO 4
• G.F. MARCATO - VIA MADONNA DELLA SALUTE 51/53
• SARTO COMPUTER - VIA ARMISTIZIO 79
Provincia di Padova
• COMPUTER SERVICE - BORGO TREVISO 150 - CITTADELLA
Treviso
• BIT 2000 - VIA BRANDOLINI D'ADDA 14
• GUERRA EGIDIO & C. - V.LE CAIROLI 95
Provincia di Treviso
• DE MARIN COMPUTERS - VIA MATTEOTTI 142 - CONEGLIANO
• SIDESTREET - VIA SALVO D'ACQUISTO 8 - MONTEBELLUNA
• FALCON ELETTROAUDIOVIDEO - VIA TERRAGGIO 116 - PREGANZIOL
Venezia
• GUERRA EGIDIO & C. - VIA BISSUOLA 20/A - MESTRE
• TELERADIO FUGA - SAN MARCO 3457
Provincia di Venezia
• GUERRA EGIDIO & C. - VIA VIZZOTTO 29 - SAN DONA' DI PIAVE
• REBEL - VIA F. CRISPI 10 - SAN DONA' DI PIAVE
Verona
• CASA DELLA RADIO - VIA CAIROLI 10
• TELESAT - VIA VASCO DE GAMA 8
• Provincia di Verona
• UBER - CP 0363(RAG.SOC. DERTA) - VIA MASCAGNI 31 -CASTEL D'AZZANO
• FERRARIN - VIA DEI MASSARI 10 - LEGNAGO
• COMPUTERS CENTER - VIA CANTORE 26 - VILLAFRANCA
Vicenza
• ELET. BISELLO - V.LE TRIESTE 427/429
• SCALCHI MARKET - VIA CA' BALBI 139
Provincia di Vicenza
• SCHIAVOTTO - VIA ZANELLA 21 - CAVAZZALE
• GUERRA E. & C. - V.LE DELLE INDUSTRIE - MONTECCHIO MAGGIORE

FRIULI VENEZIA GIULIA
Gorizia
• E.CO. ELETTRONICA - VIA F.LLI COSSAR 23
Trieste
• AVANZO GIACOMO - P.ZZA CAVANA 7
• COMPUTER SHOP - VIA P. RETI 6
• COMPUTIGI - VIA XX SETTEMBRE 51
• CTI - VIA PASCOLI 4
Udine
• MOFERT 2 - VIA LEOPARDI 21
• R.T. SISTEM UDINE - VIA L. DA VINCI 99
Provincia di Udine
• IL PUNTO ELETTRONICO - VIA VENDRAMIN 184 - LATISANA
• IDRENO MATTIUSSI &C. - VIA LICINIANA 58 - MARTIGNACCO

TRENTINO ALTO ADIGE
Bolzano
• COMPUTER POINT - VIA ROMA 82/A
• MATTEUCCI PRESTIGE - VIA MUSEO 54
Provincia di Bolzano
• RADIO MAIR-ELECTRO - VIA CENTRALE 70 - BRUNICO
• ELECTRO RADIO HENDRICH - VIA DELLE CORSE 106 - MERANO
• ERICH KONTSCHIEDER - PORTICI 313 - MERANO
• ELECTRO TAPPEINER - P.ZZA PRINCIPALE 90 - SILANDRO
Trento
• CRONST - VIA G. GALILEI 25
Provincia di Trento

• AL RISPARMIO - C.SO VERONA 138 - ROVERETO

LIGURIA
Genova
• ABM COMPUTER - P.ZZA DE FERRARI 24 rosso
• CAPRIOTTI G. - IA MAMIANI 4r - SAMPIERDARENA
• C.tro ELET. - VIA CHIARAVAGNA 10 R - VIA SESTRI 69R
• COM.le SOTTORIPA - VIA SOTTORIPA 115/117
• FOTOMONDIAL - VIA DEL CAMPO 3-5-9-11-13 r
• LA NASCENTE - VIA SAN LUCA 4/1
• PLAY TIME - VIA GRAMSCI 3/5/7 rosso
• RAPPR-EL - VIA BORGORATTI 23 R
Imperia
• CASTELLINO - VIA BELGRANO 44
Provincia di Imperia
• CENTRO HI-FI VIDEO - VIA DELLA REPUBBLICA 38 -SANREMO
• CASTELLINO - VIA GENOVA 48 - VENTIMIGLIA
La Spezia
• I.L. ELETTRONICA - VIA V. VENETO 123
Provincia di La Spezia
• I.L. ELETTRONICA - VIA AURELIA 299 - FORNOLA DI VEZZANO
Savona
• CASTELLINO - C.SO TARDY E BENECH 101
Provincia di Savona
• CELESIA ENZA - VIA GARIBALDI 144 - LOANO

EMILIA
Bologna
• EUROELETTRICA - VIA RANZANI 13/2
• MINNELLA ALTA FEDELTA' - VIA MAZZINI 146/2
• MORINI & FEDERICI - VIA MARCONI 28/C
• STERLINO - VIA MURRI 73/75
Provincia di Bologna
• S.C. COMPUTERS - VIA E. FERMI 4 - CASTEL SAN PIETRO
• S.P.E. INFORMATICA - VIA DI MEZZO PONENTE 385 - CREVALCORE
• ARCHIMEDE SISTEMI - VIA EMILIA 124 - S. LAZZARO DI SAVENA
Modena
• CO - EL - VIA CESARI 7
• ORSA MAGGIORE - P.ZZA MATTEOTTI 20
• VIDEO VAL WILLY COMPUTERS - VIA CANALETTO 223
Provincia di Modena
• NEW MEDIA SYSTEM - VIA ROMA 281 - SOLIERA
Parma
• BABARELLI G. - VIA B. PARENTE 14/A/B
Provincia di Parma
• PONGOLINI - VIA CAVOUR 32 - FIDENZA
Piacenza
• COMPUTER LINE - VIA G. CARDUCCI 4
• DELTA COMPUTER - VIA M. DELLA RESISTENZA 15/G

TEGGIO EMILIA
• COMPUTERLINE - VIA SAN ROCCO 10/C
• POOL SHOP - VIA EMILIA S. STEFANO 9/C
Provincia di Reggio Emilia
• MACCHIONI - VIA STATALE 467 - CASALGRANDE
ROMAGNA
Ferrara
• BUSINESS POINT - VIA CARLO MAYER 85
Forlì
• COMPUTER VIDEO CENTER - VIA CAMPO DI MARTE 122
Provincia di Forlì
• TOP BIT - VIA VENETO 12 - FORLIMPOPOLI
• COMPUTER HOUSE - V.LE TRIPOLI 193/D - RIMINI
• EASY COMPUTER - VIA LAGOMAGGIO 50 - RIMINI

REPUBBLICA S. MARINO
Ravenna
• COMPUTER HOUSE - VIA TRIESTE 134
Provincia di Ravenna
• ARGNANI - P.ZZA DELLA LIBERTA' 5/A - FAENZA
• ELECTRON INFORMATICA - VIA F.LLI CORTESI 17 - LUGO
• P.L.Z. INFORMATICA - P.ZZA SERCOGNANI 6 - FAENZA

TOSCANA
Arezzo
• DELTA SYSTEM - VIA PIAVE 13
Firenze
• ATEMA - VIA BENEDETTO MARCELLO 1a-1b
• ELETTRONICA CENTOSTELLE - VIA CENTO STELLE 5/a-b
• HELP COMPUTER - VIA DEGLI ARTISTI 15-A
• TELEINFORMATICA TOSCANA -VIA BRONZINO 36
Provincia di Firenze
• WAR GAMES - VIA R. SANZIO 126/A - EMPOLI
• NEW EVM COMPUTER - VIA DEGLI INNOCENTI 2 - FIGLINE VALDARNO
• C.tro INFOR. - VIA ZNOJMO 41 - PONTASSIEVE
• COSCI F.LLI - VIA ROMA 26 - PRATO
• BARBAGLI C. ELET. - VIA F. BONI 80 - PRATO
Grosseto
• COMPUTER SERVICE - VIA DELL'UNIONE 7
Livorno
• ETA BETA - VIA SAN FRANCESCO 30
• FUTURA 2 - VIA CAMBINI 19
Provincia di Livorno
• PUNTO ROSSO - VIA BARONTINI 28 - PIOMBINO
Provincia di Lucca
• IL COMPUTER - V.LE COLOMBO 216 - LIDO DI CAMAIORE
• SANTI VITTORIO - VIA ROMA 23 - S. ROMANO GARFAGNANA
• TOP GAMES - VIA S. ANDREA 122 - VIAREGGIO
Massa
• EURO COMPUTER - P.ZZA G. BERTAGNINI 4
Carrara
• RADIO LUCONI - VIA ROMA 24/B
Pisa
• ELECTRONIC SERVICE - VIA DELLA VECCHIA TRANVIA 10
• PUCCINI S.- CP 1199 (RAG.SOC. MAREX) - VIA C.CAMMEO 64
• TONY HI-FI - VIA CARDUCCI
Provincia di Pisa
• M.C. INFORMATICA - VIA DEL CHIESINO 4 - PONTEDERA (PI)
Pistoia
• ELECTRONIC SHOP - VIA DEGLI SCALZI 3
Provincia di Pistoia
• ZANNI &C. - C.SO ROMA 45 - MONTECATINI T.

Siena
• R. BROGI - P.ZZA GRAMSCI 28
• VIDEO MOVIE - VIA GARIBALDI 17
Provincia di Siena
• ELETTRONICA di BIFOLCHI - VIA DI GRACCIANO NEL CORSO 111 - MONTEPULCIANO

LAZIO
• CENTRO INF. - D.R.R. srl - TEL. 06-5565672

UMBRIA
Perugia
• MIGLIORATI - VIA S. ERCOLANO 3-10
Provincia di Perugia
• COMPUTER STUDIO'S - VIA IV NOVEMBRE 18/A - BASTIA UMBRA
• WARE - VIA DEI CASCERI 31 - CITTA' DI CASTELLO
Terni
• CGS SOFTWARE HOUSE - VIA DONIZETTI 71/A

BASILICATA
Matera
• G. GAUDIANO ELECTRONICS - VIA ROMA ang. XX SETTEMBRE 1

PUGLIA
Bari
• ARTEL - VIA GUIDO D'ORSO 9
• COMPUTER'S ARTS - V.LE MEUCCI 12/B
• PAULICELLI S. & F. - VIA FANELLI 231/C
Provincia di Bari
• F. FAGGELLA - C.SO GARIBALDI 15 - BARLETTA
• G.FAGGELLA - P.ZZA D'ARAGONA 62A - BARLETTA
• LONUZZO G. - VIA NIZZA 21 - CASTELLANA
• TECNOUFF. - VIA RICASOLI 54 - MONOPOLI
• TANGORRA N. - C.SO V.EMANUELE 130/B - TRIGGIANO
Brindisi
• MARANGI E NICCOLI - VIA PROV. SAN VITO 165
Provincia di Brindisi
• MILONE G. - VIA S.F. D'ASSISI 219 - FRANCAVILLA FONTANA
Foggia
• BOTTICELLI G. - VIA SAV.POLLICE 2
• E.C.I. COMPUTER - VIA ISONZO 28
• LA TORRE - V.LE MICHELANGELO 185
Provincia di Foggia
• IL DISCOBOLO - VIA T. SOLIS 15 - SAN SEVERO
Lecce
• BIT - VIA 95 REGG.NTO FANTERIA 87/89
Provincia di Lecce
• TECNO UFFICIO - P.ZZA GIOVANNI XXIII 10 - GALLIPOLI
• CEDOK INFORMATICA - VIA UMBERTO I 116 - TRICASE
Taranto
• ELETTROJOLLY C.tro - VIA DE CESARE 13
• TEA - TEC. ELET. AV. - VIA R. ELENA 101

CAMPANIA
Provincia di Avellino
• FLIP FLOP - VIA APPIA 68 - ATRIPALDA
Benevento
• E.CO. INF. - VIA PEPICELLI 21/25
Caserta
• ENTRY POINT - VIA COLOMBO 31
• O.P.C. - VIA G. M. BOSCO 24
Provincia di Caserta
• M.P. COMPUTER - VIA NAPOLI 30 - MADDALONI
• DAMIANO - C.SO V. EMANUELE 23 -ORTA DI ATELLA
• FUSCO B. - VIA NAPOLI 24 - VAIRANO PATERNORA (FRAZ. VAIRANO SCALO)
• LINEA CONTABILE - VIA OSPEDALE 72/76 - SESSA A. (CE)
Napoli
• BABY TOYS - VIA CISTERNA DELL'OLIO 5/BIS
• CASA MUSICALE RUGGIERO - P.ZZA GARIBALDI 74 (INT. STAZ. F.F. S.S.)
• C.tro ELET. CAMPANO - VIA EPOMEO 121
• CI.AN - GALLERIA VANVITELLI 32
• CINE NAPOLI - VIA S. LUCIA 93/95
• DARVIN - CALATA SAN MARCO 26
• GIANCAR 2 - P.ZZA GARIBALDI 37
• ODORINO - L.GO LALA 22 A-B
• R.2 - VIA F. CILEA 285
• SAGMAR - VIA S. LUCIA 140
• TOP VIDEO - TOP COMPUTER - VIA S. ANNA DEI LOMBARDI 12
• VIDEOFOTOMARKET - VIA S. BRIGIDA 19
Provincia di Napoli
• ELECTRONIC DAY - VIA DELLE PUGLIE 17 - CASORIA
• TUFANO - S.S. SANNITICA 87 KM 7 - CASORIA
• SOF SUD - V.LE EUROPA 59 - CASTEL/MARE DI STABIA
• ELETTRONICA 2000 - C.SO DURANTE 40 - FRATTAMAGGIORE
• SPADARO - VIA ROMANI 93 - MADONNA DELL'ARCO
• GATEWAY - VIA NAPOLI 68 - MUGNANO
• VISPINI & DI VUOLO - VIA A.ROSSI 4 - POMPEI
• SPY CASH & CARRY - P.ZZA ARENELLA 6/A - NAPOLI
• NUOVA INFORMATICA SHOP - VIA LIBERTA' 185/191 PORTICI
• BASIC COMPUTER - C.SO GARIBALDI 34 - POZZUOLI
• V.C. - C.SO SECONDIGLIANO 562/B - SECONDIGLIANO
• F. ELETTRONICA - VIA SARNO 102 - STRIANO
• TECNO - VIA V. VENETO 48 - TORRE DEL GRECO
Salerno
• COMPUMARKET - VIA BELVEDERE 35
• COMPUTER MARKET - C.SO VITTORIO EMANUELE 23
Provincia di Salerno
• KING COMPUTER - VIA OLEVANO 56 - BATTIPAGLIA
• DIMER POINT - V.LE AMENDOLA 36 - EBOLI
• IACUZIO F. - VIA MUNICIPIO 14 - MERCATO SAN SEVERINO
• COMPUTER SERVICE - VIA L.DA VINCI 81 - SCAFATI

CALABRIA
Catanzaro
• C. & G. COMPUTER - VIA F. ACRI 28
• PAONE S. & F. - VIA F. ACRI 93/99
Provincia di Catanzaro
• COMPUTER HOUSE - VIA BOLOGNA (L.GO OSPEDALE) - CROTONE
• RIOLO F.LLI - VIA VENEZIA 1/7 - CROTONE
• ING. FUSTO S. - C.SO NICOTERA 99 - LAMEZIA TERME
Cosenza
• MAISON DE L'INFORMATIQUE - VIA PASQUALE ROSSI 34/C
• SIRANGELO COMP. - VIA N. PARISIO 25
Provincia di Cosenza
• HI-FI ALFANO G. - VIA BALDACCHINI 109 - AMANTIA
• ELIGIO ANNICCHIARICO &C. - VIA ROMA 21 - CASTROVILLARI
• ALFA COMPUTER - VIA NAZIONALE 341/A - CORIGLIANO SCALO
REGGIO CALABRIA
• CONTROL SYSTEM - VIA S.F. DA PAOLA 49 D
• SYSTEM HOU. - VIA FIUME ang. PALESTINO 1
Provincia di Reggio Calabria
• COMPUTER SHOP - V.LE MATTEOTTI 36/38 - LOCRI
• PICIEFFE - C.SO F. S. ALESSIO 19 - TAURIANOVA
SICILIA
• CENTRO INF. - ITALSOFT SRL - TEL. 0935-696090